

UNIVERSIDAD POLITÉCNICA DE CARTAGENA
Escuela Técnica Superior de Ingeniería de
Telecomunicación

AUTOMAN-TUNEL
Desarrollo de un Sistema de
Visión y Control para el diseño
de un Prototipo de Sistema
Automático para Aplicación de
Herbicida en Tareas de
Mantenimiento de Carreteras

Alumno: **Antonio Martínez Sánchez**
Titulación: **Ingeniería de Telecomunicación**
Directores: **José Carlos Fernández Andrés**
Pedro Javier Navarro Lorente

Cartagena, 19 de Septiembre de 2006



*A mis Padres
y a Mari Cruz*

“Preferiría comprender una sola cosa que ser el Rey de Persia”

Demócrito de Abdera

ÍNDICE

1. INTRODUCCIÓN	1
1.1. Objetivos	3
1.2 Fases del Proyecto	4
1.3. Organización de la Memoria y los Apéndices	4
2. SISTEMAS BASADOS EN VISIÓN ARTIFICIAL	5
2.1. El Problema de la Visión	5
2.2. Terminología y Campos relacionados	7
2.3. Etapas del Proceso Realizado por los Sistemas de Visión	8
2.4. Componentes de un Sistema de Visión.....	10
2.5. Campos de Aplicación de los Sistemas de Visión	10
3. SISTEMAS DE CONTROL BASADOS EN AUTÓMATAS PROGRAMABLES ..	12
3.1. Sistemas de Control	13
3.2. Automatismos Cableados y Programables	15
3.3. El Autómata Programable	15
3.4. Control por Ordenador	16
4. DESCRIPCIÓN DE LOS ELEMENTOS DEL SISTEMA	18
4.1. Túnel de Viento	19
4.1.1. Estructura.....	20
4.1.2. Sistema de Inyección	21
4.1.3 Sistema de Ventilación	22
4.1.4. Anemómetro	23
4.2. La cámara, su Óptica y el Sistema de iluminación	24
4.2.1. Sistema de iluminación	24
4.2.2. Óptica de la cámara.....	26
4.2.3. Cámara de video	29
4.2.4. Posicionamiento de la cámara.....	31
4.3. Adquisición de Imágenes	32
4.3.1. Comunicación entre Cámara y Tarjeta de Adquisición.....	32
4.3.2. Tarjeta de Adquisición.....	33
4.4. Procesamiento de Imágenes	35
4.4.1. Comunicación Tarjeta de Adquisición – Procesador	36
4.4.2. El Procesador	37
4.4.3. El Sistema Operativo	38

4.5. Comunicación entre el sistema de Visión y el Autómata.....	38
4.5.1. Módulo de comunicación del PC	39
4.5.2. El Servidor OPC	40
4.5.3. Protocolo de Comunicación	43
4.6. El Autómata.....	46
4.6.1. La CPU del Autómata.....	47
4.6.2. Entradas y Salidas del Autómata	48
4.7. Sistema de Escaneado	52
4.7.1. Actuador Neumático	52
4.7.2. Terminal de Válvulas	54
4.7.3. Sensores de Proximidad.....	56
4.7.4. Sensor de Ultrasonidos	58
4.7.5. Conexiones con el Autómata	60
5. IMPLEMENTACION DEL SOFTWARE DE VISIÓN ARTIFICIAL	62
5.1. Herramientas empleadas para el desarrollo del software de visión	62
5.1.1. Lenguaje de Programación	63
5.1.2. Librerías de Visión Artificial	64
5.1.3. El Compilador.....	66
5.2. El Espacio de Trabajo	67
5.3. Captura y Preprocesado de imágenes	68
5.3.1. Análisis del Problema	68
5.3.2. Controles ActiveMIL utilizados.....	69
5.3.3. Proceso de Adquisición y Preprocesado de Imágenes	70
5.4. Comunicación con el Autómata.....	72
5.4.1. Análisis del Problema	72
5.4.2. La Clase ServerOPC	74
5.4.3. El Proceso de Escaneado	76
5.5. Calibración	78
5.5.1. Análisis del Problema	78
5.5.2. Controles ActiveMIL utilizados.....	80
5.5.3. Proceso de Calibración	81
5.6. Obtención de Resultados	81
5.6.1. Análisis del Problema	81
5.6.2. Controles ActiveMIL utilizados.....	84
5.6.3. Etapa de Procesado a Alto Nivel.....	85
5.6.4. Obtención de los Resultados.....	87
6. IMPLEMENTACION DEL SOFTWARE DE CONTROL.....	92

6.1. Herramientas empleadas para el desarrollo del software de control.....	92
6.1.1. Entorno de Programación	92
6.1.2. Lenguaje de Programación	94
6.2. La Aplicación de Control	94
6.2.1. Análisis del Problema	94
6.2.2. Comunicación con la Aplicación de Visión y los Periféricos	96
6.2.3. Descripción de la Aplicación de Control	97
7. MANUAL DE USUARIO DEL SISTEMA	99
7.1. Instalación de Software de Visión	99
7.1.1. Requisitos Previos.....	99
7.1.2. Instalación y Ejecución	100
7.2. Instalación del Software de Control.....	101
7.2.1. Requisitos Previos.....	101
7.2.2. Instalación y Ejecución	101
7.3. Manual de Uso de la Aplicación de Visión.....	103
7.3.1. Manejo del Interfaz Gráfico	103
7.3.2. Errores.....	106
7.3.3. Generación de Imágenes para la Calibración.....	108
7.4. Manual de Uso de la Aplicación de Control	110
7.4.1. Manejo Manual del Sistema de Escaneado	111
7.4.2. Ajuste del Rango de Escaneado	111
8. RESULTADOS, CONCLUSIONES Y TRABAJOS FUTUROS.....	113
8.1. Estudio sobre un pulverizador.....	113
8.1.1. Estudio de una Medida.....	114
8.1.2. Resultados en función de la Presión de Inyección	118
8.1.3. Resultados en función del Tiempo de Promediado	120
8.1.4. Resultados en función del Viento	121
8.2. Aportaciones o Conclusiones del Proyecto	124
8.3. Trabajos Futuros	125
9. INCLUSIÓN DE LA CLASE ServerOPC EN VISUAL C++ .NET.....	128
10. ActiveMIL EN VISUAL C++ .NET.....	134
11. BIBLIOGRAFÍA	139
Recursos Web.....	140

CAPÍTULO 1

INTRODUCCIÓN

El mantenimiento de las carreteras requiere de varias operaciones que se realizan periódicamente:

- Poda de la vegetación permanente que aparece en arcenes y medianas para evitar deslumbramientos o efectos indeseables del viento, etc.
- Aplicación de herbicida sobre la vegetación a eliminar (hierba, plantas silvestres, etc).
- Limpieza de señales y elementos constructivos como paredes de túneles.

Todas estas actuaciones implican la ocupación de la calzada por equipos de trabajo de diferente naturaleza en función de la operación a efectuar, con el consiguiente riesgo tanto para los usuarios de la vía, como para los componentes de referidos equipos. En general, el riesgo es inversamente proporcional al nivel de ocupación de la calzada por un lado y al tiempo de ocupación por el otro.

Para la mayoría de estas operaciones ya existen equipos con un cierto nivel de automatización que minimizan ambos factores de riesgo [1]. Sin embargo, una operación de las citadas, la aplicación de herbicida, no está automatizada en la actualidad, realizándose de forma manual o semi-manual por operarios a pie portando sistemas de rociado.

Esta operación se realiza con periodicidad de hasta tres veces al año en las carreteras de la red viaria nacional con la intención de secar la vegetación que posteriormente será eliminada mediante siega. Esta operación resulta fundamental por varios motivos: secar la planta previamente a su siega para que no aparezcan semillas que la reproduzcan, frenar su crecimiento antes de la siega y facilitar la propia operación de siega.

La técnica usada para ello es la de pulverización de una mezcla de agua y herbicida de acción sistémica (actúa por absorción de la planta a través de sus hojas o raíces). Los principales inconvenientes son la lentitud de aplicación (3-5 km/h), la presencia de operarios a pie en la vía y la deriva que se produce en la aplicación del herbicida y que afecta tanto a la vegetación permanente como a plantaciones colindantes con la vía.

En numerosos campos de la industria y los servicios, los métodos manuales están siendo sustituidos por la utilización de sistemas automatizados. En la actualidad existen en el mercado vehículos a los que se les acoplan diversos sistemas mecánicos sobre todo para operaciones de poda y limpieza con agua, lo que minimiza los riesgos citados anteriormente. Es de gran interés para las empresas que ofrecen este tipo de servicios el contar con un sistema que pueda ofrecerlos en su totalidad a un coste competitivo; la posibilidad de automatizar la tarea de aplicación de herbicida permitiría optimizar los medios mecánicos usados (principalmente los vehículos especiales que portan los equipos) mediante el mero intercambio de aperos, de igual manera que ya se hace en las labores agrícolas.

El objetivo principal del proyecto de investigación en el que se enmarca el trabajo a realizar es aplicar la robótica, la teleoperación y la visión artificial para la automatización de la aplicación de herbicida en el mantenimiento de carreteras, afrontando una problemática novedosa cuyo desarrollo ha de responder a dos retos principales: la factibilidad estrictamente técnica del procedimiento y la obtención de velocidades de operación que la hagan competitiva.

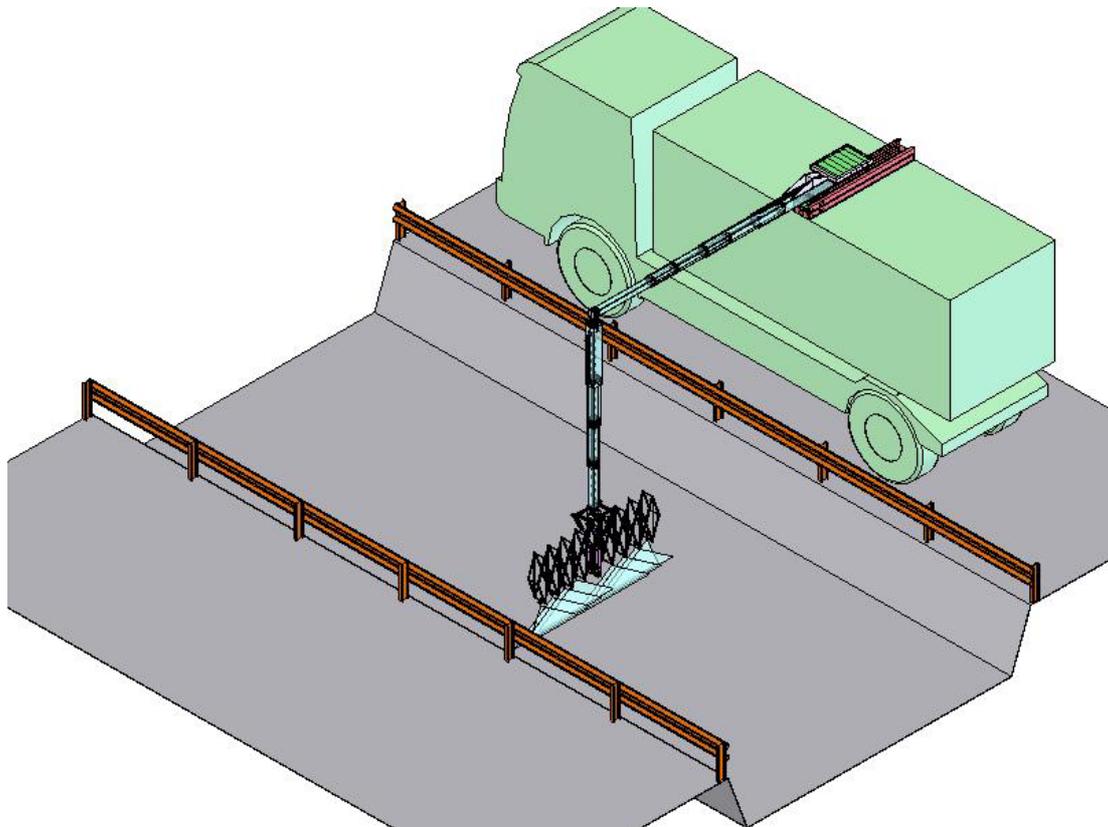


Figura 1. 1. Prototipo AUTOMAN.

Para demostrar la viabilidad de esta automatización, el grupo de investigación adscrito a la Universidad Politécnica de Cartagena DSIE (División de Sistemas de Ingeniería Electrónica) está

construyendo un primer prototipo demostrativo que cumpla con los requisitos funcionales descritos en el párrafo anterior llamado AUTOMAN, financiado por Ministerio de Industria, dentro del programa PROFIT. En el marco genérico de este proyecto de I+D es en el que se han movido las tareas de este Proyecto Final de Carrera (PFC).

1.1. Objetivos

El objetivo de este Proyecto es el desarrollo de un entorno que permita la simulación del comportamiento de AUTOMAN en un entorno controlado, de manera que permita el diseño de varios de sus parámetros.

Al sistema desarrollado en este Proyecto se le llamará AUTOMAN-TUNEL, el cual tratará de ser un sistema completo de evaluación de las prestaciones del sistema de inyección de herbicida que portará AUTOMAN. Las prestaciones del sistema de inyección comprenden la distribución y forma del chorro, o nube, de herbicida expulsado por las boquillas de pulverización, aunque el objetivo principal y que más condicionará el diseño del sistema AUTOMAN-TUNEL será la deriva producida en este chorro por el efecto del viento.

Este no se tratará por tanto de un sistema que AUTOMAN vaya a incorporar para su funcionamiento en campo, sino que servirá como banco de pruebas de laboratorio, facilitando la evaluación del comportamiento del prototipo en campo y la implementación del control del sistema de inyección de forma que éste sea autorregulable según las condiciones de viento del entorno.

Para concretar las prestaciones que AUTOMAN-TUNEL ha de ofrecer son las siguientes:

- Alojar el sistema de inyección de AUTOMAN.
- Proveer un entorno en el que se puedan fijar manualmente las condiciones de viento que actuarán sobre el sistema de inyección. Las condiciones referidas son velocidad y dirección del viento.
- Las condiciones de viento deben permanecer estables durante un ciclo de evaluación del chorro de herbicida generado por el sistema de inyección.
- Disponer de un sistema de Visión Artificial y Control que cumpla los siguientes requisitos:
 - El arranque, paro, visualización de resultados y modificación de los parámetros del sistema se realice a través de una aplicación informática que trabaje sobre entorno Windows.
 - Generar la información suficiente que permita la estimación con cierta precisión de la forma de la nube de líquido generado por el sistema de inyección de herbicida.
 - La información generada también debe proveer información para estimar la distribución espacial del promedio de flujo de herbicida.

En este proyecto también se va a realizar un estudio de verificación del sistema AUTOMAN-TUNEL, el cual estará compuesto por una serie de medidas realizadas con este sistema y que

permitan validar el correcto comportamiento de éste. Este estudio se limitará simplemente a corroborar los resultados que se obtienen con evidencia físicas para así demostrar la consistencia de estos, de este estudio no se extraerán parámetros de diseño del sistema AUTOMAN ya que esto requeriría un estudio a más largo plazo y que comprendiera mayor número de simulaciones.

1.2 Fases del Proyecto

El desarrollo del proyecto se ha cubierto con las siguientes etapas:

- Estudio de otros sistemas de Visión y de Control previamente desarrollados por el grupo de investigación DSIE, en cuyo seno se realizará este Proyecto.
- Selección de los dispositivos que se utilizarán para la implementación de los distintos subsistemas del prototipo.
- Realización del sistema AUTOMAN-TUNEL, instalación, conexión y configuración de todos sus componentes, y desarrollo de la aplicación de Visión Artificial.
- Estudio de validación del comportamiento del sistema AUTOMAN-TUNEL.
- Redacción de la memoria del Proyecto.
- Defensa pública de los resultados obtenidos.

1.3. Organización de la Memoria y los Apéndices

El resto de la memoria se organiza como se indica a continuación:

- **Capítulo 2:** Breve introducción a los Sistemas basados en Visión Artificial.
- **Capítulo 3:** Breve introducción a los Sistemas de Control con PLC.
- **Capítulo 4:** Descripción de los elementos físicos que componen el sistema AUTOMAN-TUNEL.
- **Capítulo 5:** Descripción de la arquitectura y funcionalidad de la aplicación del sistema de visión de AUTMAN-TUNEL.
- **Capítulo 6:** Arquitectura y funcionalidad de la aplicación que controla es sistema de escaneado.
- **Capítulo 7:** Manuales de usuario del sistema.
- **Capítulo 8:** Resultados, conclusiones y trabajos futuros.
- Bibliografía.
- Esta memoria se acompaña de los siguientes Apéndices útiles para quien vaya ampliar este Proyecto, trabajar con partes del Proyecto o en otros proyectos que se muevan en entornos similares:
 - **Apéndice I:** Inclusión de la clase ServerOPC en proyectos de Visual C++ .NET.
 - **Apéndice II:** Trabajar con controles ActiveMIL en proyectos Visual C++ .NET.

CAPÍTULO 2

SISTEMAS BASADOS EN VISIÓN ARTIFICIAL

La visión artificial, puede definirse como el proceso de extracción de la información del mundo físico a partir de imágenes, utilizando para ello un computador [4]. Los sistemas basados en visión artificial, pueden también considerarse sistemas autónomos capaces de simular algunas de las tareas realizadas por el sistema de visión humano (o animal) [4]. En este capítulo se describen, muy brevemente, la dificultad que entraña el problema de la visión, las características principales de este tipo de sistemas, los procesos que realizan, los elementos que los constituyen y los campos en los que este tipo de sistemas encuentran aplicación.

2.1. El Problema de la Visión

La visión es, sin duda, el sentido humano más poderoso y a la vez más complejo. En un principio se pensó que crear un sistema de visión por computador era algo bastante fácil. El argumento para ello era que los computadores eran muy potentes. Por ejemplo, aunque resolver un sistema de ecuaciones diferenciales es difícil para los humanos, los computadores lo hacen con apenas dificultad. De esta forma, si se considera una tarea trivial para los humanos, como la visión, para los ordenadores debería (hipotéticamente) ser más fácil aún. Cuando se intentaron los primeros sistemas de visión artificiales se fracasó rotundamente. La explicación de este fracaso reside en lo superficial y errado del razonamiento anterior: mientras que los humanos son conscientes de la mayoría de las etapas de procesamiento involucradas en la resolución de un sistema de ecuaciones diferenciales y, por tanto, son conscientes de la complejidad del proceso (pudiendo por consiguiente trasladarlo a un computador), la mayoría del procesamiento que se realiza en la percepción visual se lleva a cabo de forma inconsciente, siendo por ello imposible de imitar o transportar a un sistema informático. A continuación se describen algunos de los motivos que contribuyen a la complejidad del problema de la visión.

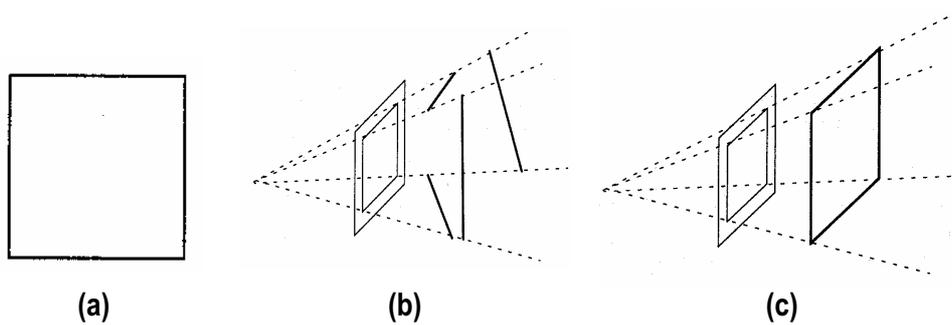


Figura 2. 1. (a) Imagen ambigua. (b) y (c) Dos escenas que darían lugar a la imagen (a). La escena (c) sería la interpretación humana de (a).

Pueden existir infinitas escenas que se proyectan en una misma imagen. Por tanto, dada una imagen es imposible determinar de forma unívoca la escena que ha dado lugar a ella. La figura 2.1 ilustra esta situación. La escena correspondiente a la imagen de la figura 2.1a puede ser el rectángulo de la figura 2.1c, o bien, algo tan diferente como los cuatro segmentos que se muestran en la figura 2.1b.

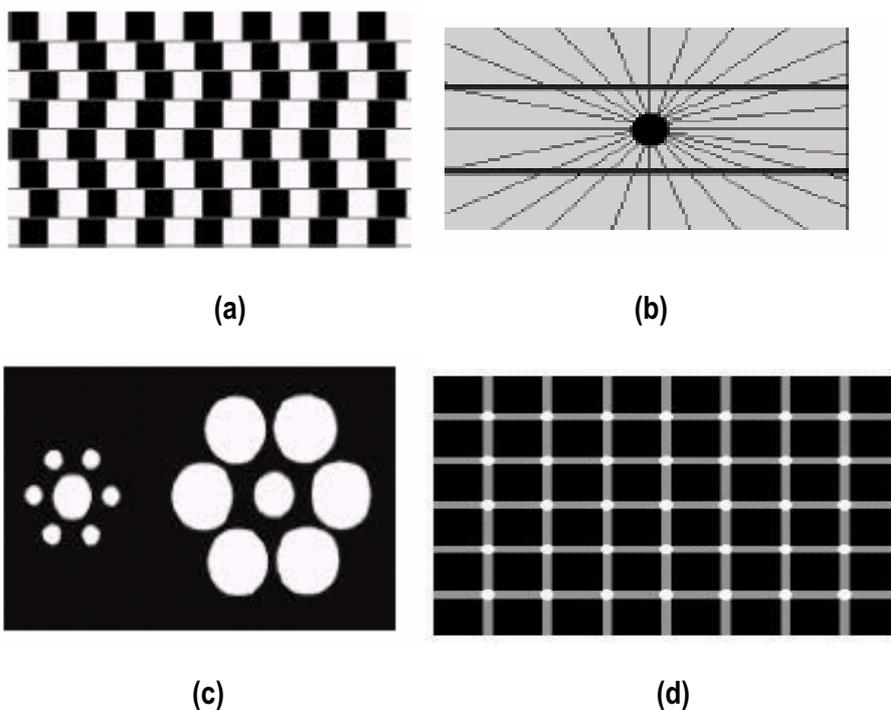


Figura 2. 2. Algunas ilusiones visuales. (a) y (b) Aunque no lo parece, las líneas horizontales son todas rectas y paralelas entre sí. (c) Los dos círculos del centro son de igual tamaño. (d) No debe haber engaño el dibujo no está vivo.

Obsérvese que la imagen de la figura 2.1a, en principio, no da la impresión de ambigüedad, es decir, los humanos no perciben todas las posibles escenas. Parece ser que en el proceso de percepción, o bien se emplea algún tipo de información de alto nivel sobre el mundo real que permite interpretar imágenes de una forma inequívoca, o bien se utilizan restricciones que permiten descartar interpretaciones. Parece existir evidencia biológica de que se emplean ambas estrategias, si bien la segunda de ellas, que se produce a un nivel inferior en el proceso de visión, proporciona una mayor capacidad de discriminación.

Estas consideraciones empíricas sugieren que, para construir un sistema de visión artificial es necesario conocer y emplear restricciones físicas sobre los objetos del mundo real y sobre su proyección en imágenes. Es conveniente destacar que esta predisposición natural a imitar al humano puede acarrear ciertas desventajas. Un ejemplo de ello son las ampliamente conocidas ilusiones visuales.

Otra dificultad a la hora de construir un sistema de visión artificial reside en la propia naturaleza de las imágenes manejadas por el computador. En primer lugar, éstas son digitales, lo que sin duda añade una dificultad adicional y, además, están afectadas por ruido diverso. En segundo lugar, y más importante aún, en los niveles de intensidad de los píxeles intervienen un gran número de factores como son, fundamentalmente:

- La iluminación de la escena.
- La geometría del objeto.
- El color y la textura de las superficies.
- Los parámetros de la cámara.

Lo realmente complicado de esto, es que la contribución individual de estos factores es imposible de precisar, con lo cual se complica el proceso de extracción de información de los objetos de la escena.

Finalmente, destacar que el volumen de información a procesar en una imagen de tamaño normal es considerable. Por ejemplo, una imagen monocroma de 256 niveles de intensidad de 512x512 (262.144 píxeles) son 0.25Mbytes, lo que significa que realizar operaciones sobre las imágenes puede ser muy costoso.

2.2. Terminología y Campos relacionados

Existen varios términos que hacen referencia al campo de la visión artificial: visión por computador, visión de máquina, visión de robot, visión computacional, análisis de imágenes, interpretación de escena. Cada uno de estos términos, aunque abordando el mismo problema, posee connotaciones distintas y enfatizan aspectos distintos involucrados en el proceso de visión artificial.

Por ejemplo, el término “visión máquina” es más utilizado en ingeniería y aplicaciones de ésta, haciendo referencia a la “construcción” de máquinas capaces de percibir e interpretar el entorno.

El término “visión computacional” surgió de la investigación de psicofísicos, técnicos en computación, neurólogos, etc. El objetivo de la visión computacional es expresar el proceso de la visión en términos de computación, entendiéndola no sólo como computación numérica sino en un sentido más amplio como sistema de procesamiento algorítmico abstracto.

El término “visión robot” hace referencia a aquellos métodos y algoritmos particularmente diseñados para dotar a un robot (normalmente manipulador) de percepción visual de su entorno. Una característica de estos sistemas proviene, precisamente, de las restricciones que este entorno impone. No se contemplan, por ejemplo, escenarios naturales, y sí es habitual el control en la iluminación.

Existen varios campos muy estrechamente relacionados en la visión por computador:

- El *procesamiento de imágenes* que involucra la transformación de una imagen para obtener otra de más calidad o mejor acondicionada para posterior extracción de la información.
- Los *gráficos por computador*, donde se aborda el problema de plasmar en un formato bidimensional el mundo real. Este proceso es el inverso al que se realiza en la visión artificial.
- El *reconocimiento de patrones*, que aborda la clasificación de objetos en clases representadas por prototipos y patrones.
- La *inteligencia artificial*, y más concretamente los problemas de interpretación, aprendizaje y razonamiento cognitivo.

Si se considera el espacio definido por estas cuatro componentes se puede decir que la visión artificial se sitúa en algún lugar de este espacio. Dependiendo del problema concreto, una componente puede ser más importante que otra. Así, por ejemplo, en un problema estéreo normalmente se involucra menos Inteligencia Artificial que en un problema de identificación de la escena. En un problema de inspección automática no es relevante, normalmente, la relación geométrica entre el mundo 3D y la imagen, mientras que son de vital importancia el procesamiento de la imagen, y el reconocimiento de las regiones y formas que en ésta aparecen.

2.3. Etapas del Proceso Realizado por los Sistemas de Visión

De manera general, el proceso realizado por los sistemas basados en Visión Artificial se puede dividir en varias etapas [4], tal y como se muestra en la figura 2.3.

- **Adquisición de imágenes:** tiene por objeto plasmar el mundo real (tridimensional) en una imagen digital (bidimensional).
- **Preprocesamiento:** incluye aquellas operaciones encaminadas a preparar la imagen para procesamientos posteriores, por ejemplo: la eliminación de ruido, el realce, etc.
- **Detección de bordes:** su importancia es crucial en muchas aplicaciones de visión ya que permite extraer los bordes de los objetos que aparecen en la imagen, facilitando su posterior segmentación y caracterización.
- **Segmentación:** tiene por objeto identificar y separar los objetos de interés del resto de la imagen. La identificación suele realizarse seleccionando aquellas regiones que exhiben unas ciertas características de interés (color, textura, intensidad, etc). Posteriormente, los objetos se separan entre sí y del fondo agrupando píxeles contiguos con características homogéneas.
- **Extracción de características:** permite describir ciertas características de los objetos. Las características analizadas en esta fase no se analizan a nivel de píxel como en el caso de la segmentación, sino que miden propiedades de los objetos como su forma, tamaño, color medio, etc.
- **Reconocimiento:** los objetos se clasifican como pertenecientes a aquella clase o prototipo cuyas características más se asemejan a las obtenidas en el paso previo.

- **Localización:** se procede a localizar al objeto en el espacio tridimensional. Para ello es necesario recurrir a técnicas de triangulación o a restricciones del espacio obtenidas a partir de cierto conocimiento a priori de la escena.
- **Interpretación:** a partir de la información obtenida en las etapas anteriores se procede a interpretar la escena, considerando para ello la relación entre los objetos simples previamente reconocidos y localizados, así como cierto conocimiento sobre restricciones y reglas que rigen el mundo real.

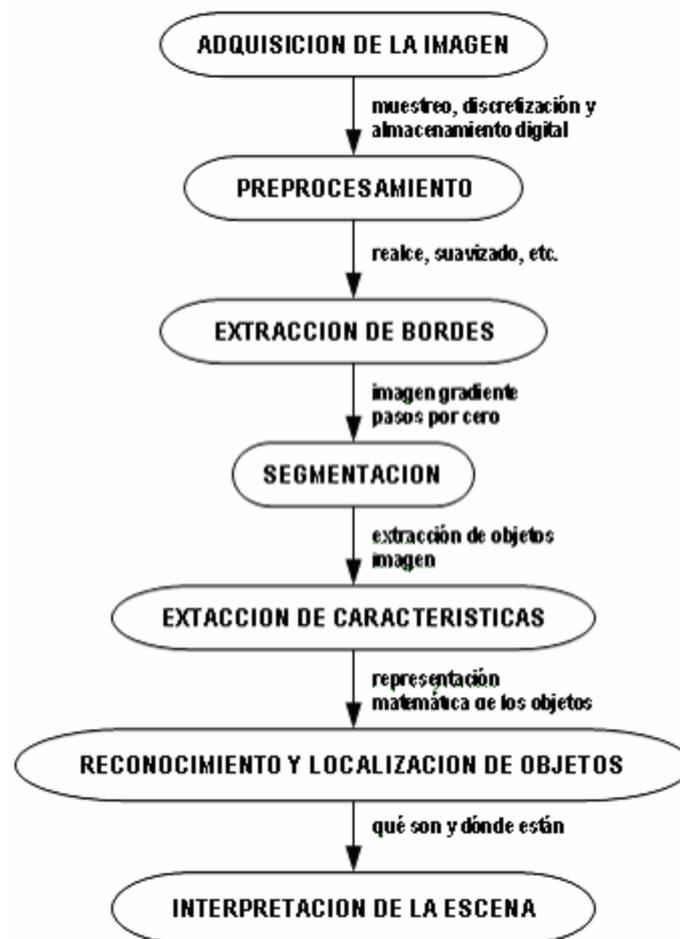


Figura 2. 3. Etapas del proceso seguido por un sistema de visión por computador.

De manera mucho más general, estas etapas suelen agruparse en dos niveles: visión de bajo nivel y visión de alto nivel. La visión de bajo nivel incluye las primeras etapas de procesamiento encaminadas a obtener características más o menos básicas de la imagen, tales como bordes, regiones, atributos de estas regiones, movimiento, etc. El análisis de la escena o visión de alto nivel, toma las características extraídas en el nivel anterior y construye una descripción de la escena a un nivel superior. Este nivel a menudo requiere de mecanismos de interpretación enormemente complejos para los que, en ocasiones, aún no se ha hallado una solución o ésta resulta enormemente costosa de calcular.

2.4. Componentes de un Sistema de Visión

Los componentes básicos de un sistema de visión [4] son los que se muestran en la figura 2.4. La entrada al sistema suele proceder de una o varias cámaras de vídeo, aunque también puede originarse en otro tipo de sensores como ecógrafos, escáneres ópticos, escáneres de resonancia magnética, etc. La información recogida por cualquiera de estos sensores es enviada (normalmente de forma analógica) a un computador donde una tarjeta de adquisición se encarga de su digitalización y en ocasiones también de su almacenamiento y procesamiento a bajo nivel. Estas tarjetas pueden ser conectadas al computador mediante los buses más comunes, en particular, bus VME, Q-bus (Sun Microsystem), bus PCI, PCI-Express, PCI-X y Un-bus (Apple Macintosh).

Las prestaciones de estas tarjetas varían tremendamente y en consecuencia también su coste. Así, podemos encontrar las que únicamente permiten la digitalización y el almacenamiento de una imagen monocromo, hasta otras que incluyen hardware para procesamiento en tiempo real de imágenes en color.

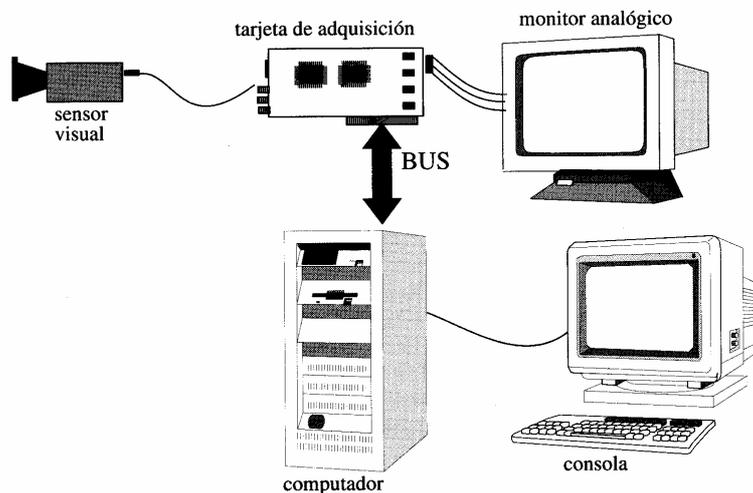


Figura 2. 4. Componentes básicos de un sistema de visión.

Para la visualización de las imágenes capturadas puede ser necesario, dependiendo del tipo de tarjeta, un monitor analógico además del monitor digital del computador. Normalmente, las imágenes son transferidas por el BUS desde la tarjeta a la RAM del computador, donde podrán terminar de ser procesadas.

2.5. Campos de Aplicación de los Sistemas de Visión

En la actualidad, la visión artificial tiene aplicación en numerosos campos tanto científicos como industriales [4]. En la siguiente tabla se recogen algunos de ellos, incluyendo un ejemplo ilustrativo de cada tipo de aplicación.

CAMPO DE APLICACIÓN	EJEMPLO
INDUSTRIA	

CAMPO DE APLICACIÓN	EJEMPLO
Control de calidad e inspección	Inspección de láminas de aluminio, nivel de llenado de botellas,...
Identificación de piezas	Clasificación automática de piezas
Ensamblaje	Montaje de chips en placas de circuito impreso
Medición de objetos	Medición del espesor granular en lingotes de hierro
Guiado de robots	Control de robot en soldadura con arco
MEDICINA	
Pruebas de laboratorio automáticas	Recuento de glóbulos en sangre, detección de células anormales,...
Diagnóstico por computador	TAC, resonancia magnética, etc.
TELE-MEDICION	
Exploración geofísica	Interpretación de fotografías aéreas
Meteorología	Pronóstico meteorológico de imágenes reobtenidas por satélites
DEFENSA MILITAR	
Vigilancia por satélite	Detección de movimiento de tropas o misiles
Guiado de larga distancia	Guiado de misiles de crucero
Armas/municiones inteligentes	Misiles aire-aire, bombas guiadas por visión
OTROS	
Campo Científico	Análisis de dinámica de flujos turbulentos, procesamiento de imágenes de telescopios
Sistemas de seguridad	Detección de movimientos, identificación intrusos,...
Entornos peligrosos	Inspección de conductos en centrales nucleares, desactivación de explosivos

Tabla 2. 1. Campos de aplicación de la visión por computador.

El sector industrial es uno de en los que más sensiblemente se aprecia el interés por este tipo de aplicaciones, sobre todo en temas relacionados con la automatización de la inspección y el control de procesos. En estos casos, los sistemas de visión aportan precisión, robustez y sin duda una reducción de costes a medio y largo plazo.

CAPÍTULO 3

SISTEMAS DE CONTROL BASADOS EN AUTÓMATAS PROGRAMABLES

Se podría definir el control como la manipulación indirecta de las magnitudes de un sistema denominado *planta* a través de un sistema llamado *sistema de control* [9]. La figura 1.1 muestra esquemáticamente un diagrama de bloques con los dos elementos esenciales: sistemas de control y planta.

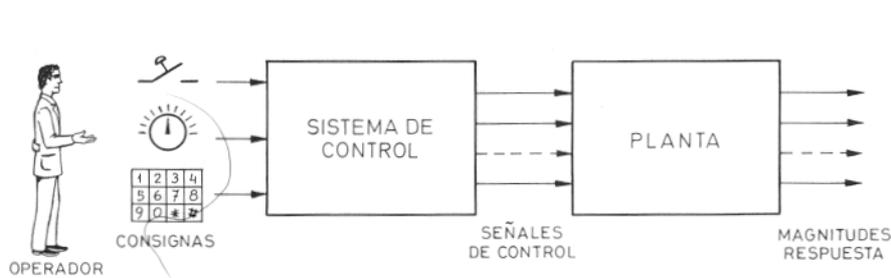


Figura 3. 1. Sistema de Control.

Los primeros sistemas de control se desarrollaron con la revolución industrial de finales de siglo XIX y principios de siglo XX. Al principio, se basaron casi exclusivamente en componentes mecánicos y electromagnéticos, básicamente engranajes, palancas, relés y pequeños motores, pero a partir de los años cincuenta empezaron a emplearse los semiconductores, que permitían el diseño de sistemas de menor tamaño y consumo, más rápidos y con menor desgaste. En la década de los setenta, la complejidad y las prestaciones de los sistemas de control se incrementaron gracias al empleo de circuitos integrados y en particular los de tipo programable (sistemas basados en microprocesadores).

Al tiempo que se desarrollaban los circuitos integrados lo hacían también los ordenadores digitales, si bien su empleo en la industria quedaba restringido al control de procesos muy complejos, debido a su elevado coste, necesidad de personal especializado para su instalación y manejo, y a la poca facilidad de interconexión (interfaz) con el proceso, donde se manejan habitualmente tensiones y corrientes fuertes, para las cuales no suele estar preparado un ordenador.

Los autómatas actuales han mejorado sus prestaciones respecto a los primeros en muchos aspectos, pero fundamentalmente a base de incorporar un juego de instrucciones más potente, mejorar la velocidad de respuesta y dotar al autómata de capacidad de comunicación. Los juegos de instrucciones incluyen actualmente, aparte de operaciones lógicas con palabras, operaciones aritméticas, tratamiento de señales analógicas, funciones de comunicación y una serie de funciones de control no disponibles en tecnología clásica de relés. En definitiva, se podría decir que los grandes autómatas actuales se acercan cada vez más a las prestaciones de un pequeño ordenador, siendo algunos incluso programables en lenguajes típicamente informáticos como el BASIC.

Sin embargo, la principal virtud del autómata sigue siendo su robustez y facilidad de interconexión al proceso y la tendencia actual no es precisamente la de acercarlo más a las prestaciones de los ordenadores en cuanto a su capacidad de cálculo, sino dotarlo de funciones específicas de control y de canales de comunicación para que pueda conectarse entre sí y a los propios ordenadores. El resultado de esta integración es la red de autómatas conectada a ordenador, capaz de ofrecer las prestaciones y ventajas de ambos sistemas al integrar en un solo sistema todas las funciones de producción asistida por ordenador (CIM).

3.1. Sistemas de Control

Según se ha introducido anteriormente, el objetivo de un sistema de control es el de gobernar la respuesta de una *planta*, sin que el operador intervenga directamente sobre sus elementos de salida. Dicho operador manipula únicamente las magnitudes denominadas de *consigna* y el sistema de control se encarga de gobernar dicha salida a través de los *accionamientos*. El concepto lleva de alguna forma implícito que el sistema de control opera, en general, con magnitudes de baja potencia, llamadas genéricamente señales, y gobierna unos accionamientos que son los que realmente modulan la potencia entregada a la planta. Esta idea se refleja en la figura 3.2.

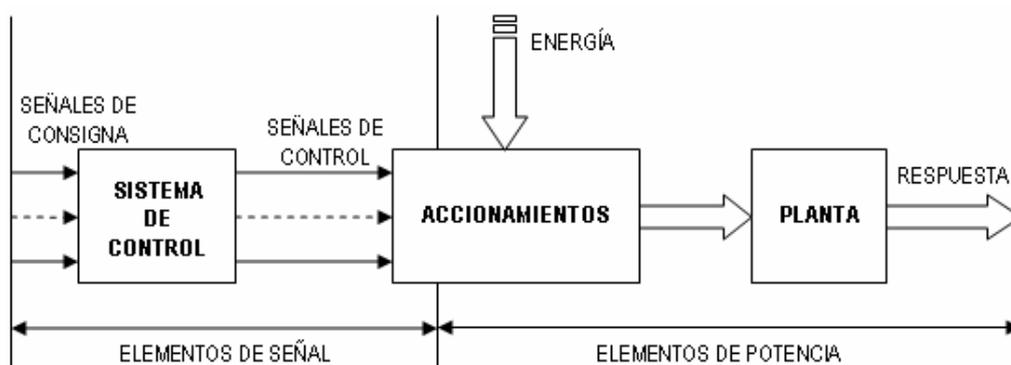


Figura 3. 2. Sistema de control en lazo abierto.

Según la definición anterior, el conjunto de sistema de control y accionamientos se limitaría a ser un convertidor amplificador de potencia que ejecuta las órdenes dadas a través de las magnitudes de consigna. Este tipo de sistema de control se denomina *en lazo abierto*, por el hecho que no recibe ningún tipo de información del comportamiento de la planta. Lo habitual, sin embargo, es que el sistema de control se encargue de la toma de ciertas decisiones ante determinados comportamientos de la planta, hablándose entonces de *sistemas automáticos de control*. Para ello se requiere la existencia de unos *sensores* que detecten el comportamiento de dicha planta y de unas *interfaces* para adaptar las señales de los sensores a las entradas del sistema de control. El diagrama de bloques será, en este caso, el de la figura 3.3. Este tipo de sistemas se denominan en lazo cerrado, ya que su diagrama muestra claramente una estructura con una *cadena directa* y un retorno o *realimentación*, formando un lazo de control.

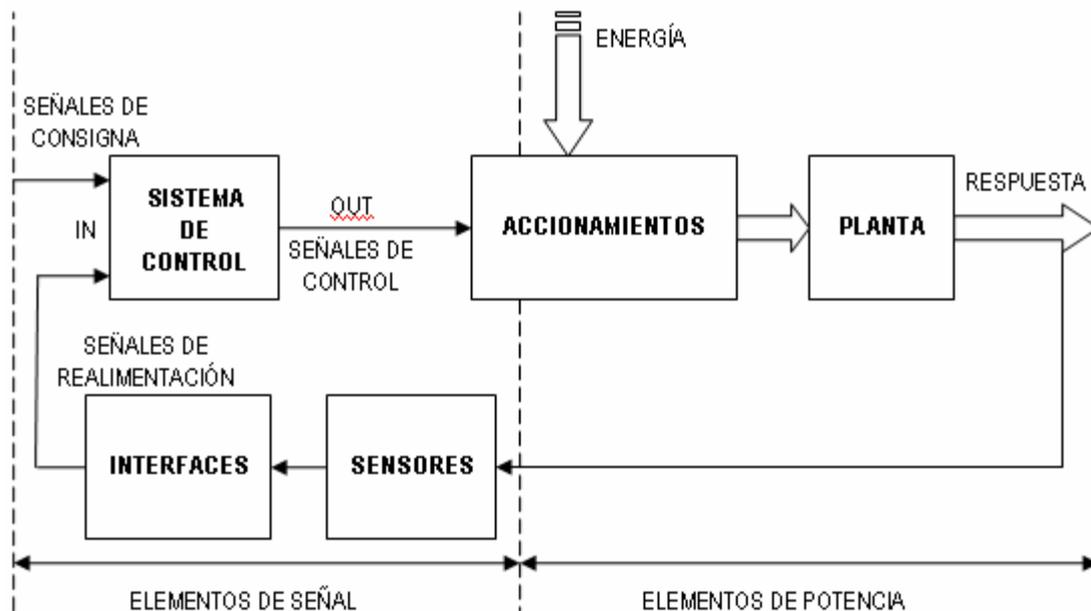


Figura 3. 3. Sistema de control en lazo cerrado.

Así pues, en el caso más general, podremos dividir el sistema de control en los siguientes bloques:

- Unidad de control.
- Accionamientos.
- Sensores.
- Interfaces.

Cabe indicar aquí que el papel del autómatas programable dentro del sistema de control es el de unidad de control, aunque suele incluir también, totalmente o en parte, las interfaces con las señales de proceso.

Al conjunto de señales de consigna y de realimentación que entran a la unidad de control se les denomina genéricamente *entradas* y al conjunto de señales de control obtenidas *salidas*. Según la naturaleza de las señales que intervienen en el proceso, los sistemas de control pueden dividirse en los siguientes grupos:

- Sistemas analógicos.
- Sistemas digitales.
- Sistemas híbridos analógico – digitales.

3.2. Automatismos Cableados y Programables

Una de las claves del éxito de los autómatas programables frente a los equipos de réles, o incluso frente a equipos construidos a base de circuitos integrados, ha sido la posibilidad de realizar funciones muy diversas con un mismo equipo (hardware estándar) y cambiando únicamente un programa (software). Atendiendo a este criterio se pueden clasificar los sistemas de control en dos grandes grupos:

- Sistemas cableados (poco adaptables).
- Sistemas programables (muy adaptables).

Los primeros realizan una función de control fija, que depende de los componentes que lo forman y de la forma en que se han interconectado. Por tanto, la única forma de alterar la función de control es modificando sus componentes o la forma de interconectarlos. Los sistemas de control programables, en cambio, pueden realizar distintas funciones de control sin alterar su configuración física, sino sólo cambiando el programa de control.

Tratándose este capítulo sobre autómatas programables, estas definiciones deben matizarse algo más, puesto que, estrictamente hablando, cualquier equipo basado en un microprocesador es en principio programable, pero para ello se requiere personal altamente especializado y equipos de desarrollo de cierta complejidad. En definitiva, del atributo “programable” se beneficia en este caso el fabricante del equipo, para el cual supone que con un hardware estándar se puede variar dentro de ciertos límites la función del equipo; pero normalmente no está en la mano del usuario el poder alterar sus funciones, por lo que para éste último el equipo es “de programa fijo” o “adaptado a medida”:

3.3. El Autómata Programable

A lo largo de los primeros párrafos de la introducción y a lo largo del apartado de sistemas de control se han ido clasificando los sistemas de control según diferentes criterios, al tiempo que se iban situando a los autómatas programables dentro de cada una de estas clasificaciones. Se está, pues, en condiciones de dar una descripción de que se entiende por autómata programable.

Desde el punto de vista de su papel dentro del sistema de control, se ha dicho que el autómata programable es la unidad de control, incluyendo total o parcialmente las interfaces con las señales de proceso. Por otro lado, se trata de un sistema con un hardware estándar, con capacidad de conexión directa a las señales de campo (niveles de tensión y corriente industriales, transductores y periféricos electrónicos) y programable por el usuario.

Al conjunto de señales de consigna y de realimentación que entran en el autómata se les denomina genéricamente *entradas* y al conjunto de señales de control obtenidas *salidas*,

pudiendo ser ambas analógicas o digitales. Así pues, existen autómatas compactos que incluyen una unidad de control y un mínimo de entradas y salidas, y luego tienen previstas una serie de unidades de expansión que les permiten llegar hasta 128 o 256 entradas/salidas. Para aplicaciones más complejas se dispone de autómatas montados en rack con posibilidad de hasta unas 2000 entradas/salidas controladas por una única unidad central (CPU).

Existe también la posibilidad, en autómatas grandes, de elección entre varios tipos de CPU, adaptados a la tarea que deba realizarse o incluso de múltiples CPU trabajando en paralelo en tareas distintas. Así, las posibilidades de elección, tanto en capacidad de proceso como en números de entradas/salidas, son muy amplias y esto permite afirmar que se dispone siempre de un hardware estándar adaptado a cualquier necesidad.

Esta adaptabilidad ha progresado últimamente hacia el concepto de *inteligencia distribuida*, gracias a las comunicaciones entre autómatas y a las redes automática – ordenador. Esta técnica sustituye el gran autómata, con muchas entradas/salidas controladas por una única CPU, por varios autómatas, con un número menor de E/S, conectados en red y controlando cada punto o sección de una planta bajo el control de una CPU central.

CARACTERÍSTICA	AUTÓMATA ÚNICO	INTELIGENCIA DISTRIBUIDA
Capacidad de procesamiento	Buena	Óptima
Estructuración en bloques	Buena	Óptima
Facilidad de mantenimiento	Buena	Óptima
Almacenajes de mantenimiento	Altos	Menores
Disponibilidad sistema en averías locales	Baja	Alta
Cableado	Grande	Reducido
Modularidad	Poca	Mucha
Coste de la instalación	Óptimo	Bueno
Posibilidades de modificación y ampliación	Buenas	Óptimas
Acceso a recursos compartidos	Rápido	Más lento
Rapidez de procesamiento	Buena	Óptima

Tabla 3. 1. Comparación de los sistemas de inteligencia distribuida frente al autómata único.

3.4. Control por Ordenador

Algunos procesos complejos requieren sistemas de control con una gran capacidad de cálculo, conexión a estaciones gráficas, múltiples canales de comunicación, facilidad de adaptación, capacidad de multiproceso, etc. Para ellos se han venido utilizando miniordenadores a los que se han adaptado interfaces específicas para la planta a controlar. Actualmente esta solución no está descartada, pero resulta económicamente cara y poco estándar, sobre todo por el hecho de que el ordenador no suele disponer de interfaces adecuadas para recoger y enviar las señales de planta.

Se ha de considerar, además, que la frontera entre un autómata de gama alta y un ordenador es cada vez más difusa, ya que dichos autómatas incorporan funciones de cálculo potentes,

capacidad de programación en alto nivel, herramientas de gestión de la programación,..., y, por otro lado, permiten fácilmente comunicarse entre sí o con un ordenador central. Así pues, la tendencia actual en el control de procesos complejos es utilizar los autómatas en red o como periféricos de un ordenador, con lo cual se combinan la potencia de cálculo del ordenador y la facilidad de interfaces estándar que ofrece el autómata. El sistema de control resultante de esta combinación ofrece las siguientes prestaciones:

- Sistema programable con una gran potencia de cálculo.
- Gran cantidad de software estándar para manipulación de datos y gestión de la producción.
- Interfaces estándar de ordenador para estaciones gráficas, utilizadas para monitorizar los procesos.
- Control descentralizado con inteligencia distribuida, sin interrumpir todo el proceso cuando hay fallos del control central.
- Sistema de comunicación estándar LAN o WAN.
- Posibilidad de acceso al sistema a través de Internet.
- Facilidad de interfaz con la planta.
- Mantenimiento fácil por secciones.
- Disponibilidad de herramientas de test y mantenimiento.
- Posibilidad de visualizar el proceso en tiempo real.
- Programación fácil a nivel de secciones.
- Flexibilidad para realizar cambios.

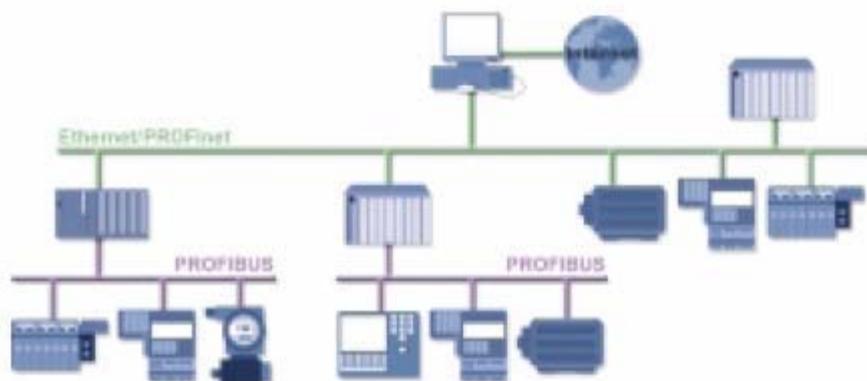


Figura 3. 4. Sistema de control compuesto por una red distribuida de autómatas conectados entre sí y con un PC a través de estándares como PROFIBUS y PROFINet.

CAPÍTULO 4

DESCRIPCIÓN DE LOS ELEMENTOS DEL SISTEMA

En este capítulo se describen en detalle todos los elementos que forman parte del sistema desarrollado en el presente Proyecto. De manera muy general, es posible identificar tres grupos de componentes: 1) el túnel de viento propiamente con su estructura, y sistemas de inyección y de ventilación, 2) el sistema de visión, y 3) el sistema de escaneado controlado por un autómata y que a su vez se comunica con el sistema de visión. Este esquema básico se recoge en la figura 4.1.

Una vez que se ha comenzado la pulverización, accionada desde el panel de control del túnel de viento y gracias al sistema de inyección que éste incorpora, ya se puede comenzar a extraer información sobre la pulverización con el sistema de visión, el cual funciona correctamente gracias al sistema de escaneado. Dentro del túnel puede haber, o no, una corriente de aire generada por el sistema de ventilación del túnel, la cual deformará más o menos el chorro generado por el pulverizador que va a ser escaneado.

La información sobre el pulverizado se genera gracias a la iluminación proporcionada por el sistema de escaneado, de manera que sobre la nube de gotas de agua se irá proyectando un haz de luz con un perfil plano y paralelo a la superficie del suelo, además este haz va a ir desplazándose gracias a un sistema neumático para así poder cubrir varios cortes la nube. Mientras, el sistema de visión a través de una cámara va registrando la información codificada en la luz reflejada por las gotas de agua generadas por el pulverizador. Esta información es enviada desde la cámara a un procesador, compuesto por un PC, a través de una tarjeta de adquisición. El procesador del sistema de visión, que está conectado con el autómata que controla el sistema de escaneado, le irá indicando al sistema de escaneado cual es la fase del proceso de escaneado en que se encuentran. Una vez recogida toda la información interesante generada durante un proceso de escaneado, ésta pasará a ser procesada con el fin de medir las derivas (o diámetro de un corte perpendicular de la nube de pulverización) máximas y mínimas, y estimar la forma de la nube resultado de la pulverización.

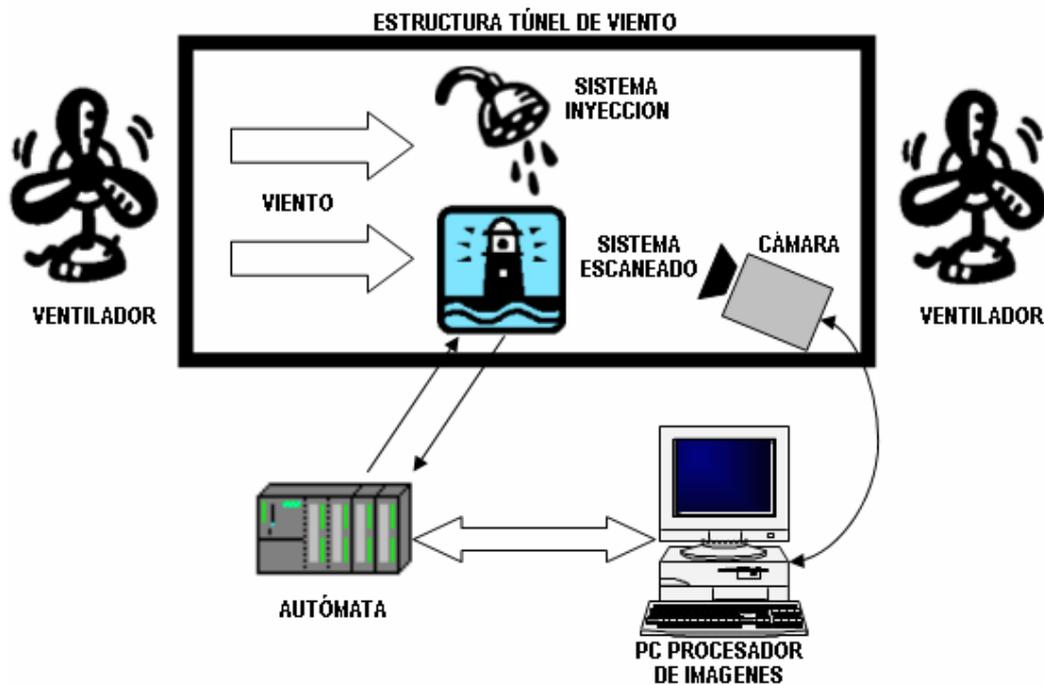


Figura 4. 1. Esquema básico del sistema AUTOMAN-TUNEL.

4.1. Túnel de Viento

El túnel de viento se compone de una estructura metálica que sirve de esqueleto para el resto de elementos del propio túnel, así como a parte de los del sistema de visión y de escaneo. Alrededor de este esqueleto metálico se acoplan los sistemas de inyección y de ventilación, controlados a su vez por un cuadro eléctrico.

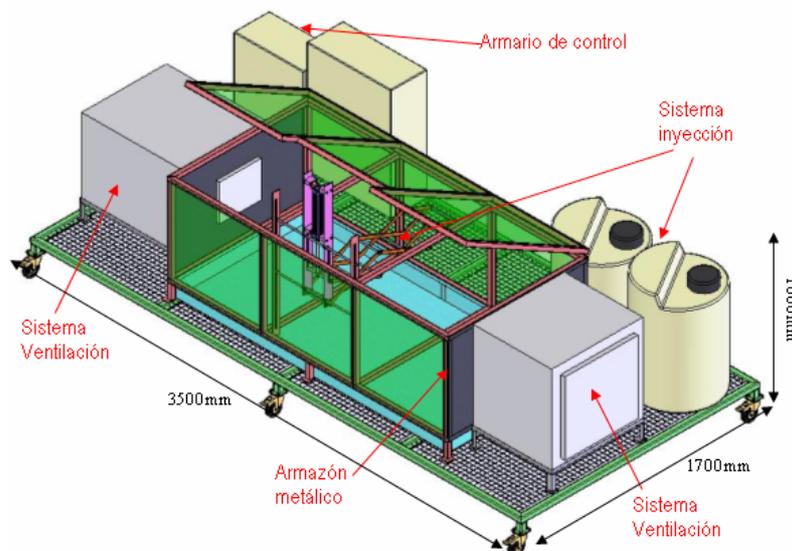


Figura 4. 2. Túnel de Viento.

4.1.1. Estructura

La base estructural del túnel de viento es simplemente un cajón de 2 metros de largo, 1 metro de ancho y 76 centímetros de alto. Este cajón, al igual que el resto de componentes del túnel, descansa sobre un suelo de Tramex gracias a unos marcos de apoyo. En la otra cara del suelo de Tramex hay una rueda en cada uno de los cuatro extremos que facilitan el transporte de túnel, y que a su vez disponen de frenos que permiten fijación del túnel (ver figura 4.3).

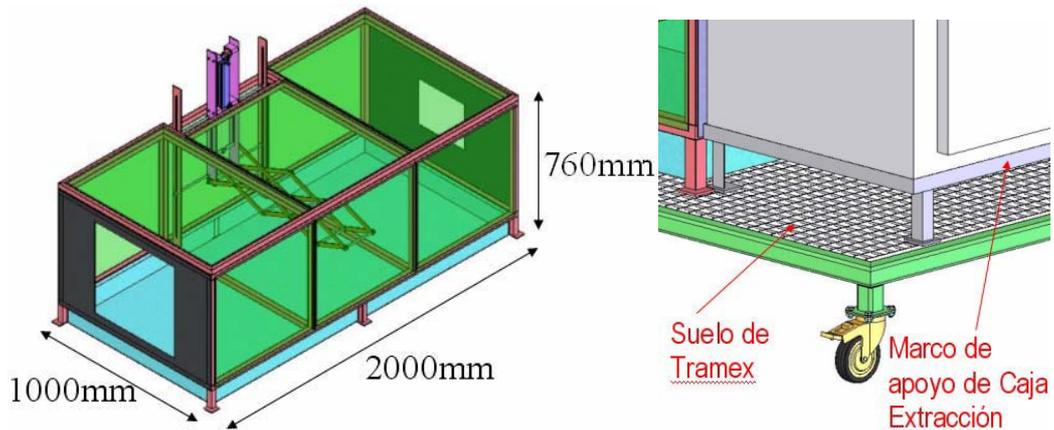


Figura 4. 3. Cajón que da forma al túnel de viento y su apoyo sobre un suelo de Tramex.

El cajón está conformado por un esqueleto metálico que se cierra con unos paneles de plástico transparente excepto en el suelo, el cual es metálico y se encargará de recoger el líquido existente en el túnel y encauzarlo hacia los bidones de almacenaje (ver figura 4.6). Debido a que para el correcto funcionamiento del sistema de visión dentro del cajón no debe entrar luz ajena a la que emite el sistema de escaneado, todo el cajón a sido forrado con cartulinas negras (lo más opacas posible), evitando así la entrada de luz exterior por los paneles transparentes (figura 4.4).

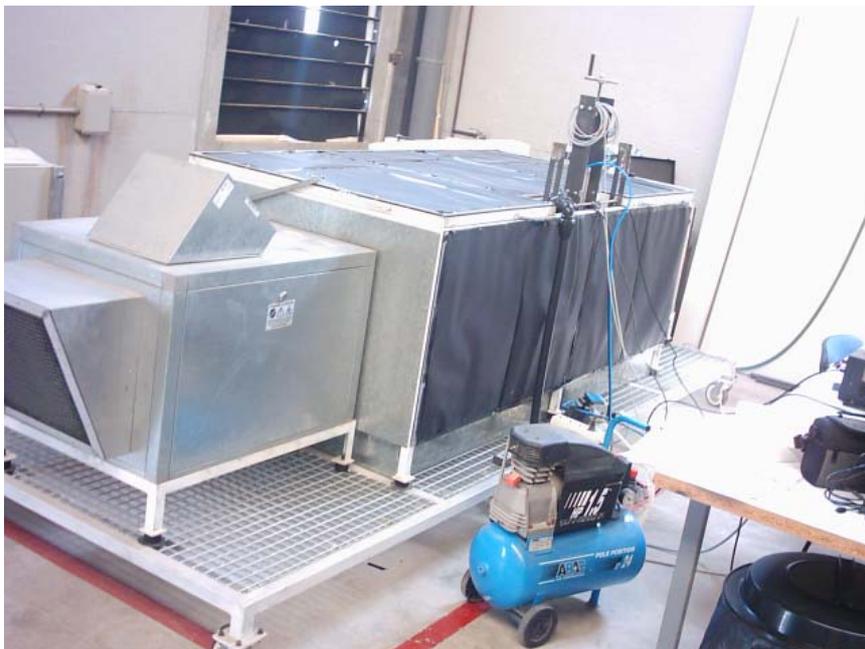


Figura 4. 4. Fotografía del túnel en donde se puede observar el aislamiento lumínico del interior del túnel.

El cajón contiene dos aperturas, una pequeña en donde se anclará la caja impulsora de viento y otra más grande donde se ancla la caja extractora de viento (figura 4.3). Dentro del cajón se encuentra una tijera anclada a uno de los lados del túnel, sobre la cual descansará el pulverizador (figura 4.5). La apertura y altura a la que sitúa la tijera pueden ser reguladas manualmente, aunque en este proyecto siempre se ha considerado que este se encuentra a una altura tal que el pulverizador siempre estará situado a 49 cm del suelo del cajón, y 51.5 cm de la cara del cajón en donde se ancla la tijera (figura 4.5).

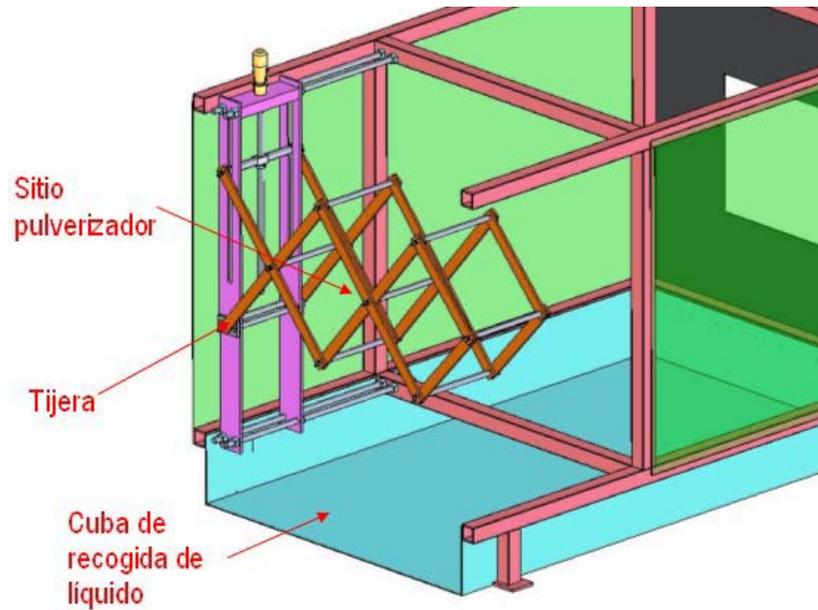


Figura 4. 5. Vista del interior del cajón y del anclaje de la tijera.

4.1.2. Sistema de Inyección

El sistema de inyección está formado por un par de depósitos en donde se almacena el líquido, y desde donde es canalizado por una serie de tuberías de goma hacia el pulverizador, la presión necesaria para mover el líquido entre las tuberías y que éste salga con presión del pulverizador la pone un motor eléctrico. Tanto el motor como los elementos de almacenaje descansan sobre el suelo de Tamex, además del cuadro eléctrico que controlará al motor eléctrico.

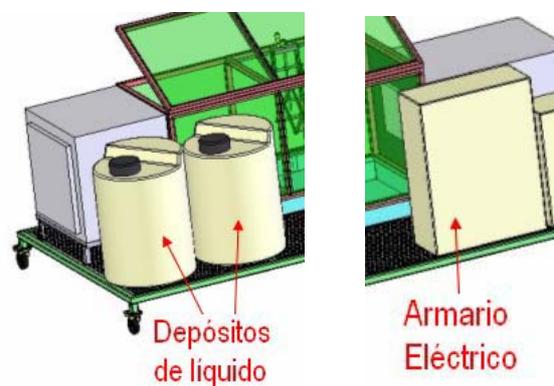


Figura 4. 6. Depósitos de líquido y cuadro eléctrico sobre el suelo de Tamex.

Las tuberías unen estos elementos entre sí y con el pulverizador, que como ya se dijo en apartado anterior está dentro del cajón, formando un circuito hidráulico donde el líquido fluye de los elementos de almacenaje hacia el pulverizador. Hay dos depósitos de líquido conectados entre sí, de tal manera que siempre hay el mismo nivel de líquido en los dos, de uno de ellos toma el líquido el motor, inyectando parte del él hacia la tubería que lleva al pulverizador, el sobrante se canaliza hacia el otro depósito. La cantidad de líquido, sobre el total absorbido por el motor, que es encaminada hacia pulverizador viene determinada por la posición de una llave, luego esta llave determina también la presión sobre el pulverizador. Esta presión puede ser medida con un manómetro alojado entre la llave y la tubería que lleva hasta el pulverizador.



Figura 4. 7. Fotografía en donde se aprecian las uniones entre el motor y el depósito mediante tuberías, además de la llave que regula el paso hacia el pulverizador situada al lado del manómetro. Al fondo se aprecia el cuadro eléctrico.

4.1.3 Sistema de Ventilación

El sistema de ventilación está formado por dos cajas situadas cada una en uno de los extremos del cajón del túnel. Cada caja contiene en su interior un ventilador accionado por un motor eléctrico. Están formadas por chapa metálica, y están completamente cerradas por 4 de sus 6 lados. En los dos lados restantes, que son opuestos entre sí, poseen aperturas para permitir el paso del viento. Por una de las aperturas se acoplarán a la apertura correspondiente del cajón, mientras la otra apunta hacia el exterior.

Una de las cajas servirá como impulsora de aire y la otra como extractora, generando así un flujo de viento sobre el cajón del túnel en dirección perpendicular a la tijera (en donde se sitúa el pulverizador). Así se simulan las condiciones de trabajo de AUTOMAN en donde, normalmente, va a haber un viento predominante en dirección perpendicular a la tijera debido al movimiento del vehículo que lo porta. La condición de impulsora o extractora viene determinada por la forma del ventilador que contienen y por el sentido de giro de éste. Los ventiladores de ambas cajas están

conectados con el cuadro de control eléctrico del túnel, el cual controla el arranque y velocidad de éstos.

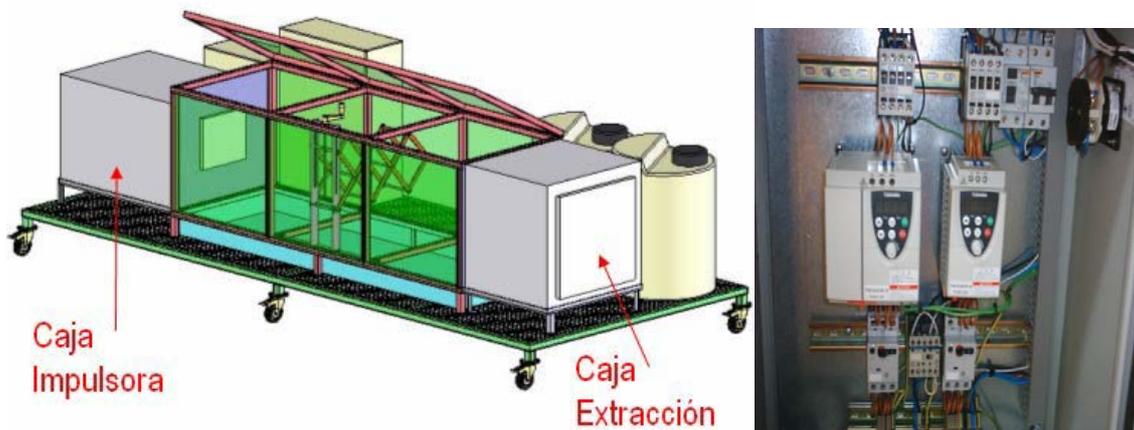


Figura 4. 8. Posición de las cajas sobre el túnel (izquierda). Interior de cuadro de control en donde se alojan los reguladores de velocidad de los ventiladores (derecha).

4.1.4. Anemómetro

Sujeto sobre la tijera y lo más cerca del pulverizador posible, pero si alterar su funcionamiento, se ha situado un anemómetro para medir la velocidad del viento que genera el sistema de ventilación. La sonda es de tipo *propeller*, es decir mediante unas aspas transforma la velocidad del viento en giros que un contador registra. Se trata de un dispositivo sencillo y cuyas características se ajustan a las necesidades del Proyecto, es robusto, fácil instalación (poco sofisticado), tamaño y coste ajustados, pero proporciona la suficiente precisión.

En concreto el modelo utilizado es el MiniAir4 de la compañía suiza Schiltknecht Masstechnik AG [Web 9]. Para visualizar los resultados se utiliza un display digital que puede realizar integraciones de 5s o 20 segundos, según la precisión que se desee, y se encuentra fuera del cajón, conectándose con la sonda a través de un cable.



Figura 4. 9. Colocación de la sonda sobre la tijera (izquierda) y display (derecha) en donde se muestra la velocidad calculada.

4.2. La cámara, su Óptica y el Sistema de iluminación

De manera genérica, para la obtención de una señal de video digital han de seguirse una serie de pasos: en primer lugar, la luz proveniente de los objetos del entorno (escena tridimensional) pasa a través de una óptica que la concentra sobre un plano; a continuación, un sensor visual, situado en el plano donde se concentra la luz, capta la intensidad luminosa en forma de señales eléctricas analógicas, que mediante la combinación de un conversor Analógico – Digital y circuitería secuencial son digitalizadas y secuenciadas dando lugar a una señal digital unidimensional que codifica la imagen.

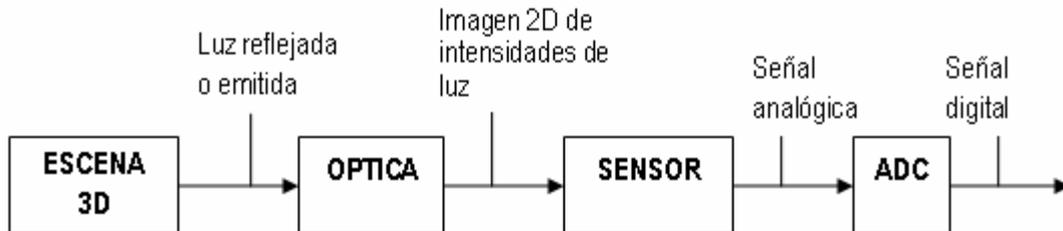


Figura 4. 10. Esquema del proceso de formación de una señal de video digital.

4.2.1. Sistema de iluminación

En visión artificial existen muchas técnicas de iluminación, entre ellas cuentan la retroiluminación, iluminación difusa, iluminación estructural, iluminación direccional. La elección de una u otra depende fuertemente del procesado que posteriormente se aplicará a las imágenes capturadas [4].

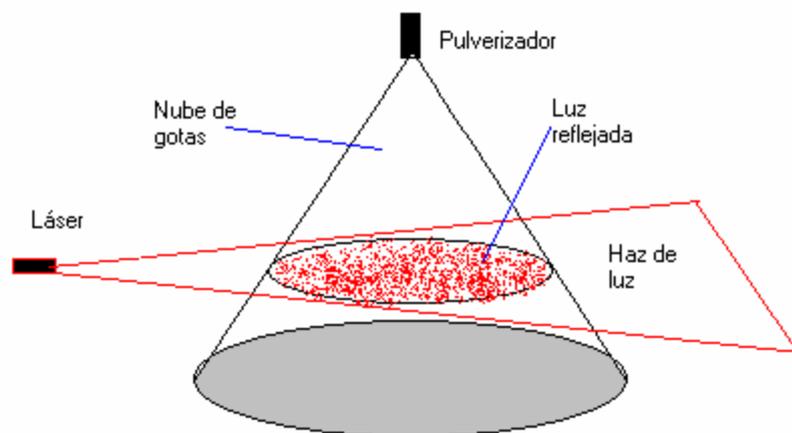


Figura 4. 11. Esquema del sistema de iluminación utilizado.

En este Proyecto se intenta obtener información de la nube de gotas generada por un pulverizador de líquido, la cual evidentemente tendrá tres dimensiones (3D). No obstante, las imágenes obtenidas por la cámara son en dos dimensiones (2D), y solo hay una cámara y además ésta no se va a mover. Lo que se va a hacer es proyectar un haz de luz en solo unas direcciones determinadas, más concretamente en un solo plano, de forma que solo las gotas que pertenezcan a este plano reflejen la luz, es decir toda la luz captada (en ausencia de luz residual) pertenecerá al mismo plano. Para hacer una reconstrucción de la escena basta con

mover el láser de iluminación, o lo que es lo mismo, el plano de los objetos sobre tantas posiciones como precisión se desee.

Todo esto funciona porque el líquido es bastante transparente, ya que el sistema se ha diseñado para trabajar principalmente con agua. Así el haz plano del láser puede atravesar toda la nube obteniéndose información de todo el corte completo de la nube. Pero debe existir un compromiso, puesto que a la vez una pequeña parte de esta luz ha de reflejarse. La reflexión de la luz sucede en cada gota ya que existe una interfase en la que cambia el medio de propagación de la luz, de aire pasa a líquido, con índices de refracción distintos, luego parte se refractará y parte se reflejará según la ley de Snell, además según una proporción gobernada por el coeficiente de reflexión [11]. Para el caso del agua este coeficiente es bajo, lo que sumado a que la reflexión es difusa debido lo aleatorio de la geometría de las gotas, implica que la cantidad de luz que alcanzará la cámara será muy poca lo que supone:

- El cajón del túnel ha de estar completamente aislado de luz exterior ajena a la del sistema de iluminación, además de minimizar la cantidad reflejada de luz del láser en las paredes del cajón. Esto se consigue mediante el forrado del cajón con cartulina opaca.
- La óptica de la cámara a de trabajar con la mayor apertura posible, a fin de dejar pasar al sensor la mayor cantidad de luz posible.
- El sensor ha de estar funcionando en ganancia máxima para poder captar el más mínimo destello de luz posible. Esto tiene el inconveniente de que aumenta bastante la cantidad de ruido de las imágenes.
- Solo se capturará la banda de color rojo puesto que es en la que emitirá más eficientemente la fuente de luz, actuando como filtro burdo del ruido y la luz exterior.
- Maximizar la eficiencia del procesado de imagen de la aplicación que controla el sistema de visión para extraer la información de la luz reflejada por las gotas de la nube, y aislarla del ruido del sensor y de la luz residual.

El sistema de iluminación no es más que un láser de semiconductor con un haz puntual que emite en color rojo, a cuya salida se sitúa una lente cilíndrica que expande el haz en un solo eje generando un haz aplanado. Debido a su construcción constituye un dispositivo barato, en contraposición con lo que sería un láser de otro material y que directamente proyectase un haz aplanado. El precio a pagar es una distribución de luz no completamente uniforme sobre el haz, pero que es asumible por el sistema.

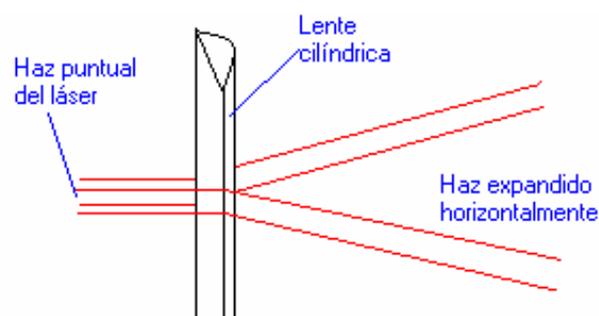


Figura 4. 12. Esquema de la fuente de iluminación donde la lente cilíndrica genera un haz aplanado a partir de uno puntual.

Esta fuente de iluminación va a limitar las dimensiones máximas de la nube de gotas, y por tanto las derivas, que se va a poder medir. El láser va a ir montado sobre el vástago del pistón, o actuador, neumático del sistema de escaneado (ver apartado 4.7.1) que no está completamente pegado a un lateral del cajón. Esto, sumado a que el haz del láser se expande con un ángulo de 33° , implica que no se va a poder medir derivas en dirección del eje del haz, aproximadamente perpendicular a la dirección del viento, superiores a 80cm (20cm menos que la longitud del cajón en esa dirección). En cuanto a las derivas perpendiculares al eje del haz, aproximadamente la dirección del viento, si se supone que la deriva máxima sucede sobre el punto nadir del pulverizador, situado a 49cm del extremo de la fuente de luz, sobre ese punto el haz se habrá expandido 53cm, luego este es el valor de la deriva máxima que se puede medir en esta dirección.

4.2.2. Óptica de la cámara

La función de la óptica de una cámara es la de captar los rayos luminosos para concentrarlos sobre el sensor de la imagen. Idealmente, la imagen obtenida debería ser una fiel reproducción de los objetos de la escena, aunque invertida y con diferente tamaño.

En este Proyecto se ha utilizado una óptica de la compañía Canon, que exteriormente adopta la forma de un cilindro metálico cuya cara anterior es una lente; en su parte posterior posee una montura en C. La montura en C es un estándar que normaliza el modo de acoplamiento entre las ópticas y las cámaras en el entorno industrial, y consiste básicamente en una simple rosca.



Figura 4. 13. Óptica de la cámara. Obsérvese la montura tipo C.

En su interior, la óptica, dispone de una agrupación de lentes de características diversas, junto con dispositivos que posibilitan el desplazamiento de las mismas variando los parámetros del sistema óptico que forman. Aunque esta agrupación de lentes se comporta, respecto de los *rayos luminosos* (término de la óptica geométrica que aproxima a la dirección de propagación de la luz) de forma parecida a como lo haría un lente simple, su utilización se justifica en aras de una mejor calidad de la imagen, compensando, en la medida de lo posible, las distintas distorsiones y aberraciones ópticas que se producen.

En la parte anterior de la óptica aparecen impresas una serie de anotaciones que definen los valores de sus principales características:

- Distancia focal de 12 mm.
- Escala graduada con las posibles distancias de trabajo, oscila entre 0.3m a ∞ .
- Diámetro de apertura con valor F de 1.4 a 1.6.

La distancia focal está relacionada con el aumento de la óptica. El factor de aumento “M” es una medida del tamaño relativo de los objetos de la imagen con relación a su tamaño real en la escena, y se define como [4]:

$$M = \frac{d_i}{d_o} = \frac{I_i}{I_o} \quad (4.1)$$

Donde d_i y d_o son, respectivamente, las dimensiones real y en la imagen de un determinado objeto capturado. Y los parámetros l_o e l_i son, respectivamente, la distancia de trabajo o de enfoque y la distancia del sensor al plano principal de la lente.

Tal y como se muestra en la figura 4.14, la distancia de enfoque l_o y la distancia focal (f) están relacionadas mediante la siguiente expresión [4]:

$$\frac{1}{f} = \frac{1}{l_o} + \frac{1}{l_i} \quad (4.2)$$

Teniendo en cuenta las ecuaciones 4.1 y 4.2, el factor de aumento y la distancia focal vienen relacionados mediante la siguiente expresión [4]:

$$f = l_o \frac{M}{M + 1} \quad (4.3)$$

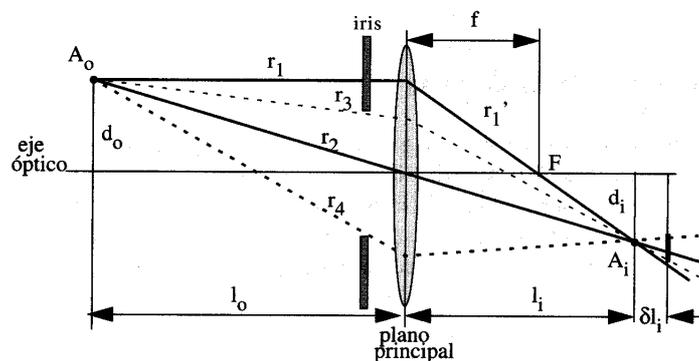


Figura 4. 14. Proceso de formación de imagen en el modelo de sistema óptico de lente delgada.

A partir de estas expresiones se puede obtener la distancia mínima de trabajo que permite cubrir el campo de visión necesario. Para este Proyecto la cámara ha de recoger información de cada uno de los planos en donde se sitúa el láser de iluminación, además también debe de cubrir la zona de trabajo del láser en cada uno de esos planos (ver apartado 4.2.1). Entonces las dimensiones del campo visual vienen limitadas en vertical por la altura del pulverizador a 49cm, y en horizontal por un poco menos que el ancho del cajón, 80cm.

El tamaño de cada píxel se conoce a partir de las especificaciones de sensor utilizado (ver apartado 4.2.3) y es de $9.9\mu\text{m}$ en sus dos dimensiones. Entonces sabiendo que el número de píxeles del array del sensor es de 656×491 , se puede aproximar el tamaño del sensor como el producto de estas dos cantidades, esto es, unos $6.5 \times 5\text{mm}$. Conocidas estas cantidades se está en disposición de calcular la distancia de trabajo mínima l_o , para la dirección más limitante del campo visual que es la horizontal:

$$M = \frac{6.5\text{mm}}{800\text{mm}} = 0.0081 \quad (4.4)$$

$$I_0 = \frac{12\text{mm}(1 + 0.0081)}{0.0081} = 148.9\text{cm} \quad (4.5)$$

Sin embargo, este resultado es válido sólo si el plano de la cámara es paralelo a la línea que forma el pulverizador con su nadir (punto más cercano del suelo del cajón), es más para lograrse la cámara debería de estar fuera del cajón. En realidad se puede jugar con la posición del plano de la cámara, aumentando así el tamaño efectivo del sensor pudiendo llegar a un máximo que se corresponde con la diagonal del sensor, esto es, 8.2mm. Ahora la distancia mínima a la que ha de situar la cámara disminuye a un valor que ya si es alcanzable manteniendo la cámara dentro del cajón:

$$M = \frac{8.2\text{mm}}{800\text{mm}} = 0.0103 \quad (4.6)$$

$$I_0 = \frac{12\text{mm}(1 + 0.0103)}{0.0103} = 117.7\text{cm} \quad (4.7)$$

No obstante, este valor solo debe tomarse como cota mínima, la situación y posición definitiva de la cámara ha dependido de muchos otros parámetros (ver apartado 4.2.4).

La distancia de trabajo no es más que la distancia, suponiendo el plano principal objeto de la óptica situado en al lente más externa, entre la cara externa de la óptica y el objeto u objetos que se pretenden capturar con la cámara. Lógicamente, esta distancia debe coincidir con la distancia de enfoque o tan próxima a ella como permita la profundidad de campo. La distancia de enfoque se puede variar manteniendo el foco fijo, gracias a un dispositivo de la óptica que permite ajustar la distancia entre el plano principal de la imagen de la óptica y el sensor de la cámara.

La profundidad de campo determina la anchura de la zona enfocada, es decir, es el rango de distancias delante y detrás del objeto que parecen estar enfocadas en el plano imagen. La profundidad de campo depende de la resolución del sensor imagen, de la apertura del diafragma y de la distancia de enfoque (ver figura 4.15).

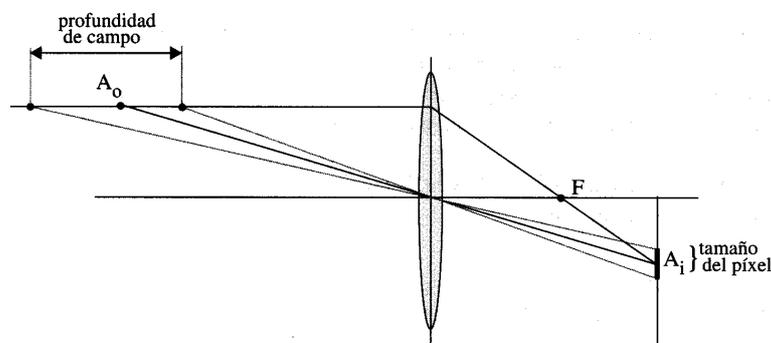


Figura 4. 15. Ilustración del concepto de profundidad de campo.

Para este caso la profundidad de campo no es un factor limitante, ya que el sistema de visión artificial va a trabajar haciendo un recuento de la luz reflejada en el líquido pulverizado, y no le interesa la forma de esas gotas. Por esto no es necesario formar imágenes nítidas de los objetos, lo realmente importante es recibir la luz que reflejan. Es más, debido a que se trabajará con líquidos transparentes (agua), a la óptica se le va a pedir que recoja la mayor cantidad de luz posible.

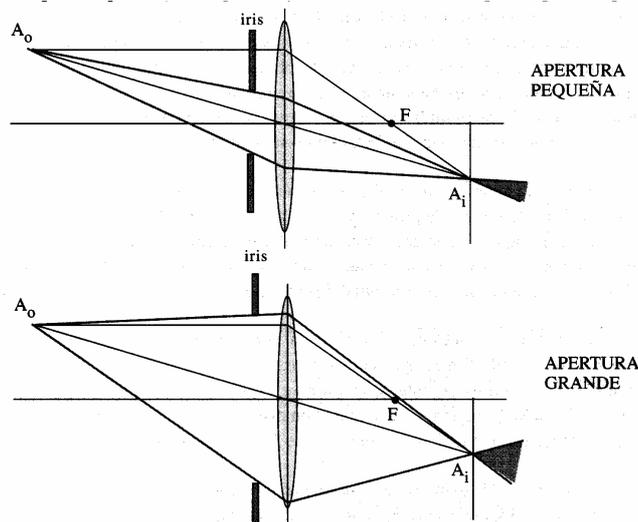


Figura 4. 16. Influencia de la apertura en la cantidad de luz recibida por el sensor.

La óptica utilizada en este Proyecto tiene una apertura regulable que oscila entre valores de F1.4 y F16. El valor de esta apertura influye directamente en la cantidad de luz que le llega al sensor y en la profundidad de campo de forma contraria (ver figura 4.16). Como la profundidad de campo no es un factor relevante en contraposición con la cantidad de luz recibida por el sensor, siempre se trabajará con valor F máximo, esto es, F16.

4.2.3. Cámara de video

Para el desarrollo de este Proyecto se ha optado por emplear la cámara modelo A601fc-2 [Web 1] de la compañía germana Basler Vision Technologies. Se trata de una cámara de bajo peso (100gr), y de tamaño reducido, sus dimensiones son 67.3x44x29mm, lo que ha facilitado su instalación. Posee el interfaz de video digital IEEE 1394 (*FireWire*) que permite aprovechar las ventajas de este estándar, así como asegurar la conectividad con la tarjeta de adquisición utilizada (ver apartado 4.3.2). Debido a su interfaz digital se puede realizar un control sencillo de parámetros de captura como por ejemplo los tiempos de exposición, tasas de captura y tamaño de captura. Se alimenta con un voltaje fijo que recibe a través del conector IEEE 1394. Además posee una montura tipo C que permite acoplarla a la óptica utilizada.



Figura 4. 17. Cámara Basler A601fc-2.

ESPECIFICACIÓN	VALOR
Tamaño Sensor (HxV Píxeles)	656x491(mono),656x490(color)
Tipo de Sensor	CMOS de escaneo progresivo
Tamaño de Píxel	9.9 μ m x 9.9 μ m
Max. frames/s a máxima resolución	60(A601) frames/s
Formato de salida	Mono: 8 bits/píxel: Color: YUV4:2:2
Color/Mono	Color o Mono
Salida de video	IEEE 1394
Sincronización	Trigger externo o por bus 1394
Control de exposición	Programable vía bus 1394
Requisitos de Alimentación	De +8 a +36 VDC (nominales 12V), max. \approx 2W
Montura	Montura en C
Dimensiones	67.3mm x 44mm x 29mm
Peso	100 gramos
Certificación	CE, FCC

Tabla 4. 1. Características de cámara utilizada.

La cámara posee un sensor CMOS de la compañía Micron Technologies Inc. que le permite realizar capturas a alta velocidad, carentes del ruido que se adquiere en exposiciones largas. Gracias a esta propiedad se pueden hacer capturas rápidas con poco ruido, luego mediante procesado de imagen eliminar este ruido e integrar todas las capturas, en lugar de realizar una sola captura en donde debido a la naturaleza del problema (ver apartado 4.2.1) el ruido sería muy grande, y la relación señal a ruido (SNR) no sería satisfactoria. Además, el propio sensor dispone de una solución propietaria del fabricante llamada *TrueBit*[®] que disminuye la cantidad de ruido generada.

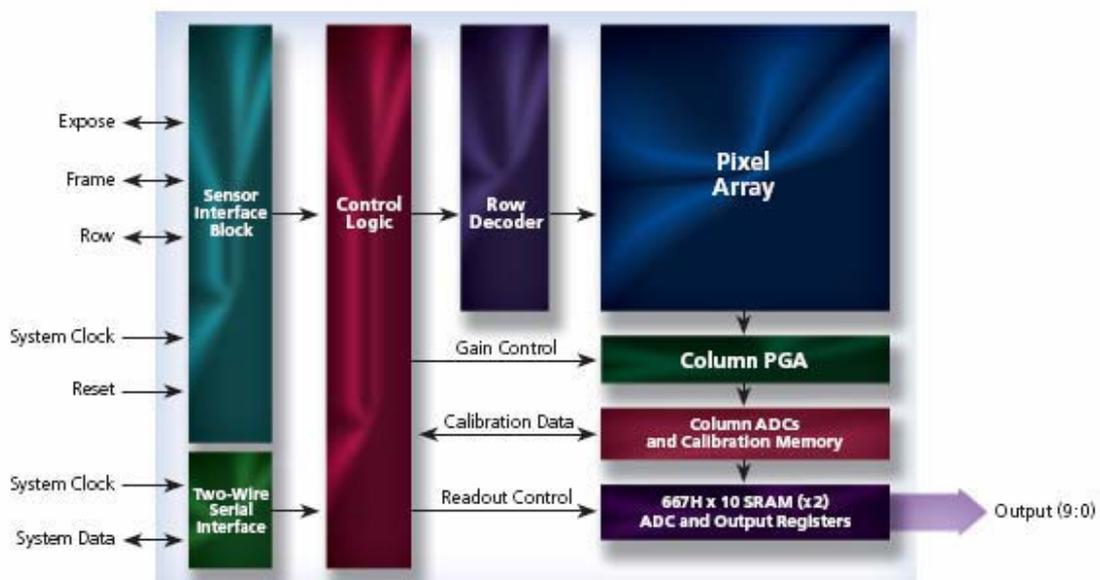


Figura 4. 18. Arquitectura interna del sensor de la cámara.

4.2.4. Posicionamiento de la cámara

La posición de la cámara condiciona en gran medida el funcionamiento de todo el sistema, por ello se ha de estudiar detenidamente dónde se va a colocar. Es más, son varios los aspectos del sistema condicionados por la posición de la cámara, por lo que conjugarlos todos exige alcanzar una solución de compromiso en la que se cumplan unos requisitos mínimos.

En principio y debido al modelo de iluminación utilizado lo ideal sería colocar la cámara sobre el pulverizador, con su plano principal paralelo a los planos con que el sistema de iluminación corte la nube de gotas de líquido. En esta posición habría garantizado una calibración de la cámara sencilla y precisa. No obstante son varios los problemas que ofrece y que hacen de esta posición inviable, en primer lugar la tijera obstruiría parte de las imágenes dificultando la extracción de resultados a la aplicación de visión artificial. El segundo problema es que la altura máxima a la que se podría colocar viene limitada por la altura del cajón, entonces capturaría imágenes de cortes situados entre 70cm y 25cm, lo que necesitaría una profundidad de campo de $2.6D$ (dioptrías). Debido al modelo de iluminación, este sistema es bastante inmune a la falta de profundidad de campo, aunque para este caso sus efectos comenzarían a hacerse visibles. El tercer y quizás mayor problema es, como ya se justificó en el apartado 4.2.2, para poder tener un campo visual suficiente para captar toda la información que permite el sistema de iluminación, la cámara habría de situarse a como mínimo 117cm del centro del plano de cualquier corte.

Las razones del párrafo anterior obligaron a desplazar la cámara hacia uno de los extremos del cajón (en donde se acoplan las cajas del sistema de ventilación). Para poder ver todos los cortes con cierta perspectiva y además cumplir con cierta holgura la restricción de distancia mínima de trabajo, la cámara debía de colocarse en la parte superior del extremo del cajón.

Entonces el problema se redujo a solo dos opciones, la cámara situada en la parte central del techo del cajón o en una de las esquinas. La segunda opción ofrece dos ventajas, en primer lugar al estar más lejos del centro de los cortes de la nube, luego cumple con mayor holgura la restricción de distancia mínima de trabajo, lo que permite no tener que girar demasiado el plano del sensor de la cámara para poder obtenerse un campo visual aceptable (ver apartado 4.2.2). En segundo lugar facilita el anclaje ya la esquina del cajón poseía ya una pieza que facilitaba el anclaje del soporte de la cámara (ver figura 4.19).

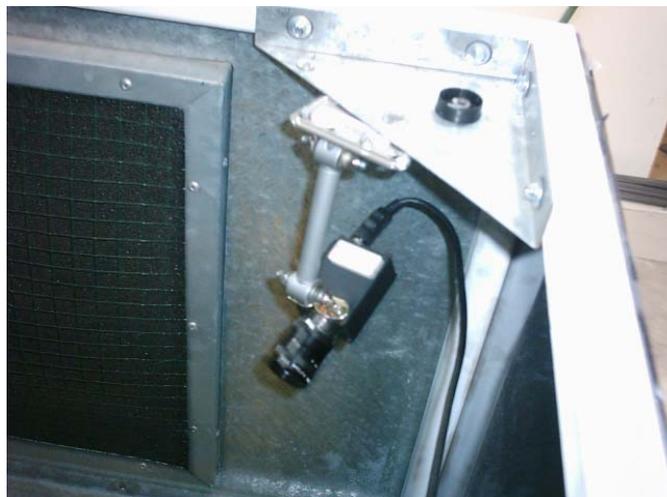


Figura 4. 19. Fotografía donde se aprecia el anclaje de la cámara a una esquina del cajón del túnel.

Más concretamente la cámara se situó en la esquina del extremo que se acopla con la caja de extracción de aire que estaba más cercana al ordenador que iba a procesar las imágenes. Esta posición cumple con los requisitos necesarios, aunque complica un poco la calibración de la cámara debido a que las imágenes sobre los cortes se toman muy en perspectiva, es decir el plano principal de la cámara está muy desalineado con los planos de corte del sistema de iluminación, sobre todo en los cortes más cercanos al pulverizador.

4.3. Adquisición de Imágenes

Como se ha visto en el apartado anterior, la cámara de vídeo empleada genera una señal de vídeo digital según el estándar IEEE 1394. Sin embargo, el sistema de visión artificial trabaja con imágenes en forma de matrices en 2D. Para liberar al procesador de esta costosa tarea se necesita de un dispositivo hardware llamado tarjeta de adquisición o frame grabber.

Una tarjeta de adquisición genera imágenes en forma de matrices 2D a partir de la señal de vídeo unidimensional que le llega proveniente de la cámara. Esta tarjeta ha de poder conectarse al interfaz de salida de la cámara, por tanto ha de ser compatible con el interfaz IEEE 1394. Además a través del conector debe alimentar la cámara según la tensión especificada.

4.3.1. Comunicación entre Cámara y Tarjeta de Adquisición

Para la comunicación y alimentación la cámara dispone del interfaz IEEE 1394 (también conocido como *Firewire*), el cual permite la transmisión de señales digitales de un gran ancho de banda como es el caso de la transmisión de vídeo en tiempo real. La distancia a la que se puede extender la comunicación si repetidores puede llegar hasta los 4.5m, lo cual es suficiente para conectar con el PC procesador de imágenes.

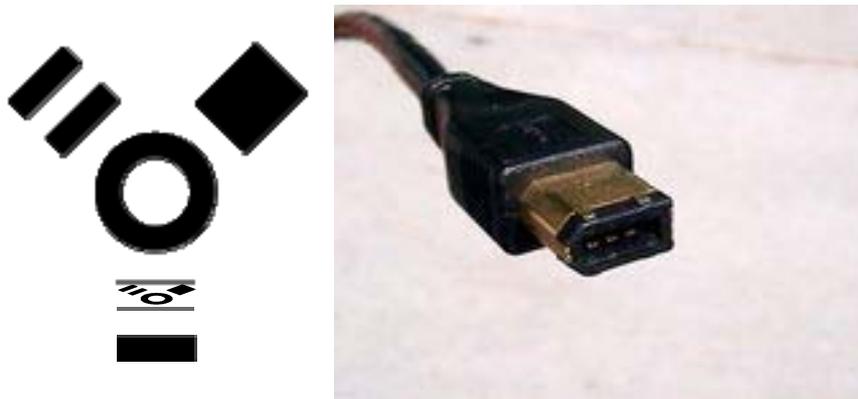


Figura 4. 20. Icono utilizado por IEEE 1394 (izquierda), fotografía del conector IEEE 1394 de 6 pines utilizado (derecha).

Este estándar se ha hecho muy popular entre los sistemas de almacenamiento de datos, videocámaras digitales, y sistemas industriales de visión artificial. En estos sistemas es usado en lugar del común USB debido a que posee una velocidad efectiva superior, mayor capacidad de distribución de alimentación, y porque no obliga a tener un PC como host, ya que permite comunicación punto a punto. Pero sobre todo porque FireWire aprovecha todas las capacidades de SCSI, y en comparación con USB 2.0 High Speed, posee mayores tasas de transferencia de datos, característica especialmente importante para los editores de vídeo.

El interfaz IEEE 1394 puede soportar transferencia de datos de 100, 200 y 400 Mbps, ofreciendo ancho de banda suficiente para las necesidades de este Proyecto. También soporta comunicación bidireccional, permitiendo controlar de forma sencilla la cámara. Además debido que existen conectores estándar para este interfaz, no ha sido necesario realizar un cableado específico para conectar la cámara con la tarjeta de adquisición. Se han utilizado conectores estándar de 6 pines en ambos extremos del cable (ver figura 4.20), debido a que éste es el conector utilizado tanto por la cámara como por la tarjeta de adquisición. Este conector lleva conexiones para alimentación, permitiendo la alimentación de la cámara a través de la tarjeta de adquisición.

4.3.2. Tarjeta de Adquisición

La tarjeta de adquisición utilizada en este Proyecto es la Meteor-II/1394 de la compañía canadiense Matrox Electronics Systems Ltd.



Figura 4. 21. Tarjeta de adquisición Matrox Meteor-II/1394.

La tarjeta Meteor-II/1394 es un adaptador de IEEE 1394 a bus PCI, a la vez que simplifica, mejora las prestaciones, y libera de tareas al microprocesador en la captura de video digital en un PC. Puesto que en este Proyecto se ha utilizado como procesador de imágenes un PC, se hace necesario el uso de esta tarjeta para poder enviar imágenes provenientes de la cámara de una manera rápida y fiable, sin lastrar las prestaciones de la aplicación de visión artificial. Además el PC utilizado no disponía de interfaz IEEE 1394 propio.

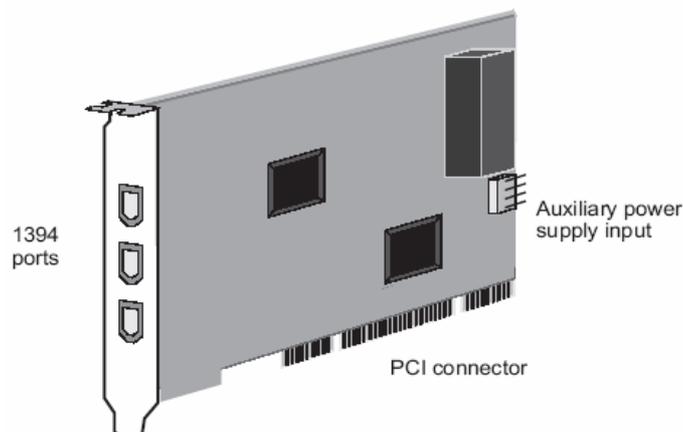


Figura 4. 22. Dibujo en el que se muestran los distintos conectores de la Meteor-II/1394.

La razón principal, sin embargo, por la que se ha optado por esta tarjeta de adquisición y que ha condicionado también la elección de la cámara, es que es compatible con Windows XP (sistema operativo utilizado por el PC sobre el que se pincha la tarjeta), y sobre todo porque permite la adquisición de imágenes y control de la cámara a partir de las librerías *ActiveMIL*. Utilizadas en el desarrollo de la aplicación de visión artificial en este Proyecto.

La Meteor-II/1394 es capaz de actuar como *bus manager* (director de ciclos), *isochronous resource manager* (gestor de flujo), y controlador de nodo (*root*). El director dispara la señal de datos con intervalos de 125 μ seg. El gestor de flujo es el responsable de reservar, distribuir, y gestionar los dos modos de flujo de datos que permite la tarjeta (isócrono y asíncrono) en el ancho de banda disponible. Como controlador *root* puede controlar todos los dispositivos periféricos conectados hasta un máximo de 62.

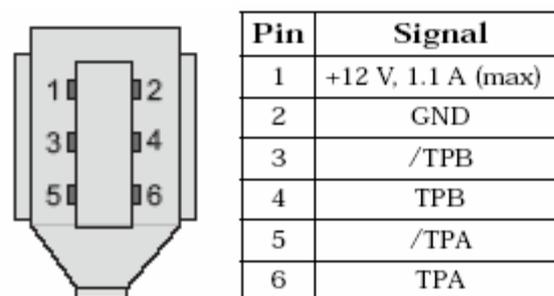


Figura 4. 23. Esquema del conector IEEE 1394 de 6 pines de la tarjeta.

La tarjeta posee tres puertos de entrada IEEE 1394 donde conectar dispositivos, generando una configuración en forma de árbol. Las tasas de datos en serie del bus de 100, 200, o 400Mbps son compartidas por los tres puertos, en este Proyecto solo se ha utilizado una cámara lo que ha permitido disponer de todo el ancho de banda (400Mbps), pudiéndose hacer capturas con toda la resolución que permitía la cámara y a alta velocidad. Los conectores de los puertos de entrada IEEE 1394 son de seis pines (ver figura 4.23), lo que le permite conectar directamente la cámara a la tarjeta de adquisición.

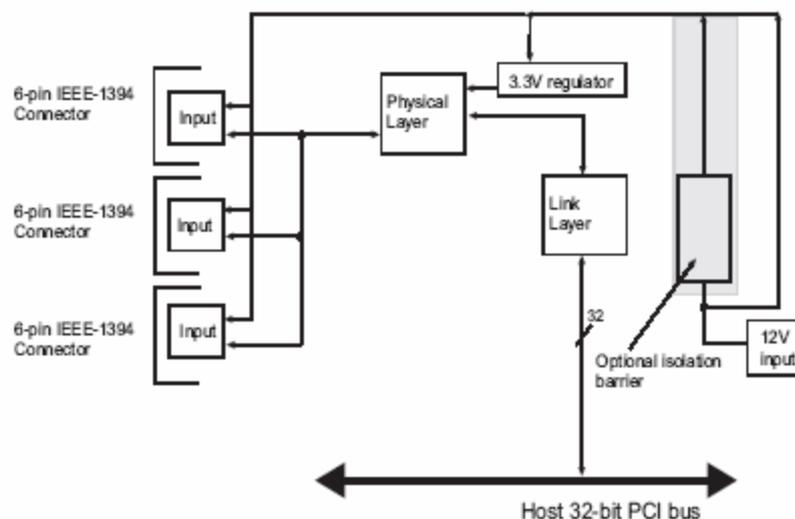


Figura 4. 24. Esquema de los módulos internos de la Meteor-II 1394.

La capa física (*Physical Layer* en la figura 4.24) es parte del interfaz bidireccional entre la capa de enlace (*Link* en la figura 4.24) y el bus serie IEEE 1394. La capa física actúa como conversor, reformateando los comandos y datos digitales que recibe, de manera que los datos puedan ser transmitidos sobre el bus serie. La capa física monitoriza las condiciones de la línea con el fin de determinar el estado de la conexión, para la iniciación y arbitraje, y para el empaquetado de los datos. La capa física impone el límite de la velocidad máxima de transmisión, funcionando hasta los 400Mbps.

La capa de enlace trabaja como interfaz bidireccional entre la capa física y el bus PCI. Este módulo es un controlador *PCI-to-1394 Host* y cumple las especificaciones PCI y 1394 OHCI. Esta capa soporta los dos modos de transferencia de datos entre nodos, asíncrono e isócrono. Cuando se transmiten paquetes, la capa de enlace envía una solicitud a la capa física, que controla la dirección de transmisión entre ambas capas. El módulo de enlace también controla la velocidad de transmisión entre él mismo y el módulo físico, y soporta transferencias de datos de 100, 200, o 400Mbps. Además, la capa de enlace es capaz de transferir datos en caché a 132Mbytes por segundo (capacidad máxima del bus PCI de 32-bit a 33MHz) sobre el bus PCI después de conectar el controlador de memoria en el *Host*. Para evitar largas latencias del bus PCI, la capa de enlace proporciona colas FIFO ocultas.

La Meteor-II/1394 también tiene una barrera galvánica aislante, usada para aislar la tierra de la tarjeta del resto de una red 1394. No obstante, en este Proyecto la red solo está compuesta por la tarjeta (conectada a un PC) y una cámara que se alimenta a través de la propia tarjeta, luego esta barrera galvánica es innecesaria. La tarjeta extrae la alimentación que enviará a la cámara a través de IEEE 1394 de la fuente del PC, a la que se conecta mediante un conector macho de 4 pines (ver figura 4.25).

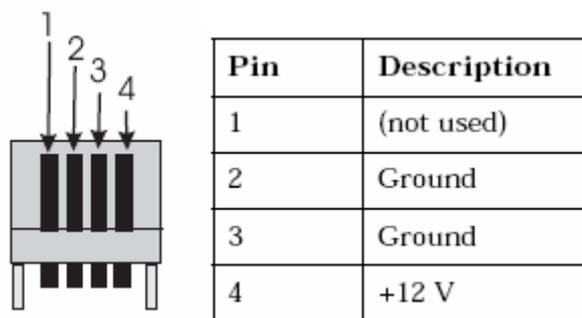


Figura 4. 25. Esquema del conector de 4 pines usado como alimentación auxiliar para IEEE 1394.

4.4. Procesamiento de Imágenes

La transformación de señales de video digital en imágenes en forma de matrices 2D, se debe a que los algoritmos de la aplicación de visión, que correrán en el procesador (microprocesador del PC), sólo son capaces de procesarlas en este formato. Una vez la imagen haya sido enviada a la memoria del procesador, éste la analizará y combinará con otras imágenes extrayendo la información interesante, esto es, las medidas de las derivas de los cortes de la nube. Pero el procesador además, debe ser consciente del estado de ejecución en que se encuentra el sistema debiendo informar al sistema de escaneado cuándo arrancar, cuándo y a cuál posición

debe moverse para realizar un nuevo corte de la nube, y cuándo se ha finalizado y debe de volver a su posición inicial.

En este Proyecto, el procesamiento de las imágenes corre a cargo de un procesador Intel® Pentium III ubicado en el mismo PC donde se aloja la tarjeta de adquisición. Este procesador es de propósito general y su arquitectura es de tipo Von Neuman. En este tipo de arquitectura, el procesador extrae las instrucciones de ejecución y los datos (imágenes en este caso) de la misma memoria. Por lo tanto, las imágenes capturadas en la tarjeta de adquisición son enviadas a la memoria del procesador, donde este es capaz de leerlas y manipularlas directamente.

La aplicación de visión artificial desarrollada para este Proyecto se ha implementado en lenguaje de programación C++. Los ejecutables que correrán en el procesador, se han generado utilizando el compilador Visual C++ .NET de Microsoft®.

4.4.1. Comunicación Tarjeta de Adquisición – Procesador

La tarjeta de adquisición posee un conector PCI de 32-bit, que se conecta directamente al bus PCI de la placa base del PC. Sin embargo, el bus PCI es de propósito no específico, por lo que también va a permitir la comunicación con el módulo PROFIBUS usado para conectar con el autómeta.

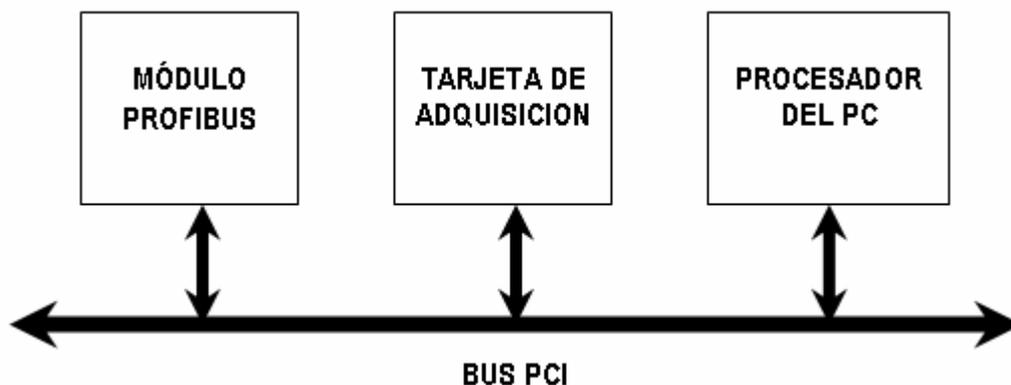


Figura 4. 26. El bus PCI es usado para la transferencia de imágenes a la memoria del procesador, y de datos entre el procesador y el módulo PROFIBUS.

PCI (*Peripheral Components Interface*) tiene una arquitectura de bus síncrono en el que todos los datos son transmitidos en función del reloj del sistema. La especificación inicial del PCI permite una velocidad máxima de reloj de 33MHz permitiendo a un bus transferir datos cada 30 nanosegundos. En revisiones posteriores de la especificación de PCI, se ha extendido la definición del bus para soportar operaciones a 66MHz o a 133MHz, sin embargo, la tarjeta Matrox-II/1394 esta preparada sólo para trabajar con buses PCI a 33MHz.

A 33MHz, y con una ranura de 32-bit como es el caso de este Proyecto, soporta una transferencia de datos máxima de 132Mbytes/seg, lo cual es suficiente para absorber las imágenes que envía la tarjeta de adquisición que recibe video digital a un máximo de 400Mbps. Los datos entre el procesador y el módulo PROFIBUS, aunque importantes, son escasos en comparación con los que enviará la tarjeta de adquisición, luego apenas robarán ancho de banda del bus PCI.

PCI soporta un riguroso mecanismo de autoconfiguración. Cada dispositivo PCI incluye un juego de registros de configuración que permiten la identificación del tipo de dispositivo (SCSI, video, *Ethernet*, etc.) y el fabricante. Otros registros permiten la configuración de las direcciones de los dispositivos de Entrada/Salida, direcciones de memoria, niveles de interrupción, etc.

Aunque PCI es usado principalmente en sistemas PC o compatibles, como es el caso de este Proyecto. La arquitectura PCI bus es independiente del procesador. La definición de las señales es genérica permitiendo que el bus PCI pueda ser usado por sistemas basados en otras familias de procesadores.

4.4.2. El Procesador

El procesador (o microprocesador) que incorpora el ordenador empleado en este Proyecto es un Pentium III de la compañía Intel®. De este modo, se le puede sacar todo el partido a las librerías de visión artificial utilizadas, ActiveMIL, que han sido optimizadas para aprovechar las ventajas que ofrecen la aceleración Intel MMX y las extensiones SIMD (instrucciones SSE que el procesador Pentium III fue el primero en incorporar).

La tecnología Intel MMX, es una extensión de la arquitectura Intel que el Pentium III lleva incorporada, y está diseñada específicamente para acelerar las aplicaciones multimedia (o parecidas). La tecnología Intel MMX está construida para manejar algoritmos computacionalmente intensivos que requieren operaciones con tipos de datos pequeños (con son los píxeles de 8-bit). La tecnología cubre varias áreas, como son las operaciones aritméticas básicas, operaciones lógicas, operaciones de desplazamiento, de comparación, e instrucciones de transferencia de datos. Estas instrucciones usan un modelo SIMD que permite al procesador con un solo cálculo realizar operaciones con 2, 4, o 8 datos mediante el empaquetado de múltiples operandos (valores de 8, 16, 32 bits) en un solo registro de 64-bit y funciones que trabajan en paralelo. En un procesador x86 compatible con la tecnología Intel MMX, las operaciones de ActiveMIL pueden ejecutarse, normalmente, 4 veces más rápido que en un procesador x86 normal. Algunas operaciones se benefician incluso más de la aceleración MMX, por ejemplo, la operación de adelgazado de imágenes puede ser hasta 16 veces más rápida.

El procesador Pentium III es de lógica programada, luego las tareas que éste ha de realizar sobre los datos, están definidas por una serie de instrucciones software almacenadas en la misma memoria que los propios datos. El microprocesador irá leyendo de la memoria el código a ejecutar, instrucción a instrucción, e irá realizando las acciones correspondientes.

Este procesador, como el de cualquier PC, ha sido diseñado para que pueda ejecutar cualquier tipo de tarea previamente programada, incluidas las relacionadas con visión artificial, si bien tendrá ciertas limitaciones en velocidad, que otros procesadores especialmente diseñados para realizar tareas de visión artificial, no tienen. Esta desventaja se compensa con la mayor facilidad que supone el programar una tarea mediante un conjunto de instrucciones software, frente al empleo de procesadores hardware específicos, cuyo uso en este Proyecto está totalmente injustificado debido a que la velocidad del procesador Pentium utilizado es suficiente, además no se trata de un aspecto crítico. Es más, el aspecto más valorado del procesador que se utiliza en este proyecto es su flexibilidad, al ser programable, y su compatibilidad con el resto de componentes.

4.4.3. El Sistema Operativo

El acceso de las aplicaciones a los recursos hardware no se realiza de manera directa, sino a través de llamadas al sistema operativo, que es en realidad el que gestiona el hardware. Así, las aplicaciones están condicionadas por el sistema operativo sobre el que se ejecutan.

En este Proyecto, se han considerado varias posibilidades con relación a qué sistema operativo emplear. En principio podría ser interesante la opción de utilizar un sistema operativo de libre distribución y abierto, como podrían ser algunas de las distribuciones de Linux, sin embargo, esta opción queda invalidada enseguida debido a su incompatibilidad con el hardware y software de Matrox® y Siemens®, utilizados respectivamente para visión artificial y control. Es por esto, por lo que el abanico de posibilidades se cierra a los únicos sistemas operativos que son compatibles con todo el hardware y software empleado, esto es, Windows XP ó Windows 2000 ambos de Microsoft®.



Figura 4. 27. El control de hardware por parte de una aplicación se realiza a través del sistema operativo.

La principal ventaja de los sistemas operativos de Microsoft® es que su uso es simple e intuitivo y su manejo está basado en ventanas, además son los sistemas operativos más utilizados para PC. Aunque las nuevas distribuciones de Linux poseen entornos bastante amigables, su uso no está tan extendido y la mayoría de usuarios, sobre todo los que poseen pocos conocimientos informáticos, desconocen su funcionamiento.

De entre las dos opciones de sistemas operativos que se podían utilizar, en este Proyecto se ha optado por Windows XP, debido a que es más avanzado que Windows 2000 y permitía la instalación de la última versión del software de Siemens® que contiene los drivers de los dispositivos empleados por este Proyecto de esta compañía, además de herramientas de diseño y configuración de sistemas de control.

Windows XP es de fácil manejo, y tiene un funcionamiento relativamente robusto, al menos lo suficiente para empeñar, de manera satisfactoria, las tareas encomendadas dentro de este Proyecto.

4.5. Comunicación entre el sistema de Visión y el Autómata

Como se ha visto, el sistema de visión obtiene y procesa imágenes digitales con el fin de obtener información sobre las derivas del corte correspondiente de la nube de líquido. La adquisición y procesamiento de imágenes corre a cargo de una tarjeta de adquisición insertada en un PC y del propio procesador del PC.

El sistema de visión debe enviar información al autómatas que controla el sistema de escaneado para informarle de que ha terminado de capturar imágenes sobre un corte, para que pase al siguiente, o que arranque o pare. Así como también debe recibir del autómatas información sobre el estado del sistema de escaneado para saber si debe o no comenzar a tomar imágenes y éstas a que corte pertenecen.

Por esto es necesario insertar en el PC un módulo de comunicación PROFIBUS que permita conectar con el autómatas mediante el protocolo estándar PROFIBUS, más concretamente PROFIBUS-DP. Es necesario recordar que quien controla realmente el sistema de escaneado es el autómatas, ya que es quien controla el estado de sus sensores y envía señales a los actuadores. A través de PROFIBUS es autómatas se limitará a informar al PC sobre el estado en que se encuentra, y el PC informará al autómatas cuando debe cambiar de estado al sistema de escaneado.

Es importante resaltar que la aplicación de visión artificial del PC no envía directamente los datos al módulo PROFIBUS y viceversa, sino que lo hace a través de un servidor OPC. Este paso intermedio que supone el apoyo en una *servidor OPC* ha supuesto una serie de ventajas, entre la que destaca el poder realizar la aplicación de visión artificial de forma totalmente independiente del autómatas utilizado.

4.5.1. Módulo de comunicación del PC

Ni la placa base del PC utilizado en este Proyecto tiene el hardware, ni Windows XP tiene el software que permite a un PC conectarse y comunicarse a través de un bus PROFIBUS. Por esta razón, ha sido necesaria la instalación de hardware como es el caso de la tarjeta CP 5611 de Siemens®, así como de la instalación del software necesario para controlarla disponible en CD que comercializa la compañía germana SIMATIC NET CD (en este Proyecto se usó la versión 6.2 con *ServicePack 1*). En este CD aparte de los drivers de la tarjeta existen varias herramientas útiles para el diseño y configuración de la red PROFIBUS, herramientas de programación, configuración y monitorización del autómatas, además de un servidor OPC.



Figura 4. 28. Aspecto de la tarjeta CP 5611.

La tarjeta PROFIBUS utilizada en este Proyecto ha sido la CP 5611. Se trata de uno de los modelos de gama baja de Siemens®, ya que por ejemplo carece de microprocesador propio. Es lógico utilizar una tarjeta de este tipo, con prestaciones limitadas, de hecho cumple con sus obligaciones dentro de este Proyecto sobradamente, de manera que no eleva el coste del sistema innecesariamente. Esta tarjeta permite la conexión del PC a un bus PROFIBUS como dispositivo PROFIBUS-DP esclavo, consiguiendo velocidades de hasta 12Mbits/s. Además, a través de esta tarjeta se pueden hacer diagnósticos de los dispositivos conectados al bus, bien sea el PC o el autómatas.

La CP 5611 ha podido ser utilizada ya que soporta el interfaz PCI, presentando las siguientes características:

- Señalizaciones con niveles de voltaje de 5 y 3.3 voltios, que ser corresponden con los niveles del estándar.
- Velocidades del reloj de 33 MHz y 66 MHz.
- Conector de 32-bit.

Para la correcta instalación del software SIMATIC NET CD 6.2+SP1 que contiene los drivers de la tarjeta, también ha sido necesaria la instalación del *ServicePack 1* de Windows XP.

4.5.2. El Servidor OPC

Como ya se ha avanzado la comunicación entre módulo PROFIBUS y la aplicación de visión artificial se realiza a través de un intermediario llamado *servidor OPC*, al que se accede a través del interfaz OPC.

OPC Foundation, creador de OPC, nació en Mayo de 1995 como resultado de la cooperación de Microsoft®, y una fuerza constituida por cinco empresas, Intellution, Opto-22, Fisher-Rosemount, Rockwell Software y Intuitiv Software. Este grupo de empresas pretendía definir una serie de especificaciones basadas en COM/DCOM y el primer borrador de las mismas fue completado al final de 1995, gracias a la colaboración de otras 90 compañías a lo largo del mundo, las cuales comprobaron estas especificaciones. El primer conjunto oficial de especificaciones se completó en Agosto de 1996.

El objetivo del comité conformado fue proporcionar una interfaz de programación de aplicaciones estándar para el intercambio de datos, que pudiese simplificar el desarrollo de drivers de I/O y mejorar el rendimiento de los sistemas de interfaz; el esquema que reinaba en ese entonces y el cual se pretendía cambiar era el siguiente:

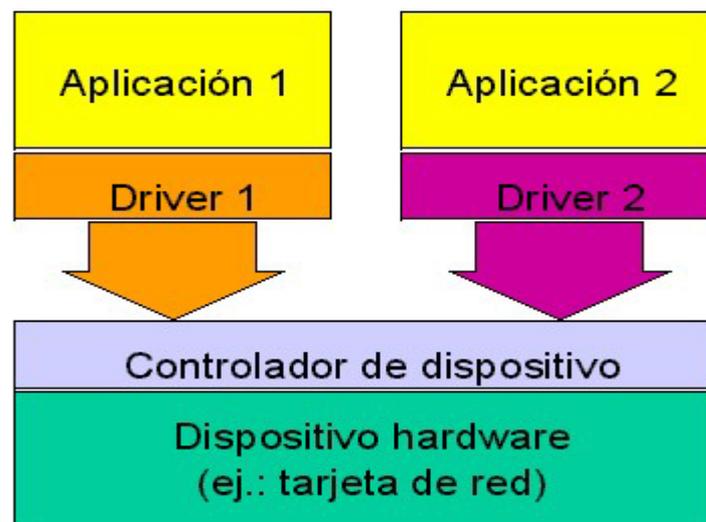


Figura 4. 29. Modelo de arquitectura de automatización industrial basado en drivers.

Según el documento *OPC Overview* publicado por *OPC Foundation* en 1998, esta arquitectura conduce a problemas como:

- Duplicación de esfuerzo: se deben escribir programas amarrados a un driver para el hardware de un vendedor particular.
- Inconsistencias entre vendedores de drivers: las características de hardware no son soportadas por todos los drivers.
- Soporte para cambios en características de hardware: un cambio en las capacidades del hardware puede ocasionar conflictos en los drivers.
- Conflictos de Acceso.

Dos paquetes, generalmente, no pueden acceder simultáneamente al mismo dispositivo, puesto que cada uno contiene drivers independientes. Los fabricantes de hardware procuran resolver estos problemas desarrollando nuevos drivers, pero son obstaculizados por diferencias en los protocolos del cliente. No se puede desarrollar un driver eficiente que pueda ser utilizado por todos los diferentes tipos de clientes.

En estas circunstancias resulta muy complejo realizar aplicaciones industriales, pues no existe una forma estándar de definir las conexiones sin tener que depender del tipo de dispositivo. OPC eliminó este problema estableciendo una interfase de comunicación común, lo cual ha beneficiado enormemente el desarrollo de aplicaciones HMI y sistemas SCADA. La siguiente figura representa una primera idea sobre un ambiente de monitoreo y control industrial basado en OPC:

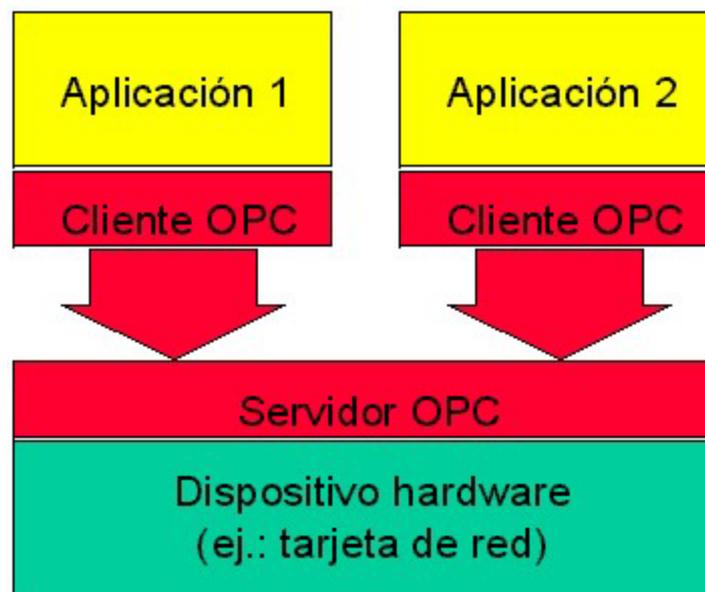


Figura 4. 30. Modelo de arquitectura de automatización basado en OPC.

OPC, (OLE para el Control de Procesos) es una especificación técnica no propietaria definida por la Entidad *OPC Foundation* y consiste básicamente en un Sistema de Interfaces Estándar basado en OLE/COM de Microsoft® [12]; con OPC es posible interoperar dispositivos industriales con sistemas de información o aplicativos de escritorio. En otras palabras, OPC permite desarrollar de una manera muy práctica y eficiente aplicaciones que pretendan comunicarse con equipos industriales controlados por PLCs.

OPC ofrece varias ventajas las cuales también fueron citadas por OPC Foundation en su OPC Overview; se destacan las siguientes:

- Los fabricantes de hardware tienen que hacer solamente un conjunto de componentes de software para que los clientes los utilicen en sus aplicaciones.
- Los desarrolladores de software no tienen que describir drivers debido a cambios en características o adiciones en un hardware.
- Los clientes tendrán más opciones con las cuales puedan desarrollar diversos sistemas de aplicación a nivel industrial.

Con OPC la integración de un sistema industrial, en un ambiente de computación heterogéneo resulta simple. Aunque en este Proyecto la arquitectura del sistema era simple, por lo cual no se expresen todas las ventajas de OPC, su uso a sentado un precedente importante que ayudará a al desarrollo de sistemas de automatización complejos que se valgan de OPC, además permitirá que se puedan ejecutar simultáneamente y compartiendo el mismo hardware, las aplicaciones de control del prototipo AUTOMAN y la aplicación de visión de AUTOMAN-TUNEL.

Posibilidades con OPC:

- Acceso a datos en línea: la lectura y la escritura eficiente de datos entre una estación central y un dispositivo de control de procesos se puede realizar de forma flexible y eficiente.
- Control de alarmas: OPC provee mecanismos para que sus clientes sean notificados de la ocurrencia de acontecimientos y de condiciones de alarmas especificadas.
- Acceso a datos históricos: OPC permite la lectura, procesamiento y corrección de datos históricos con un eficiente motor de acceso.

Con la arquitectura OPC se aprovechan las ventajas de la interfaz COM para ampliar su funcionalidad. La especificación OPC incluye lo siguiente:

- Interfase COM/DCOM para ser usada por clientes Locales o Remotos.
- Referencias a la Interfase de Automatización OLE.

En este Proyecto se cumplen todos los requerimientos de funcionalidad que permiten la utilización de OPC. Esto se debe a que la especificación de OPC contiene a Windows XP con Service Pack 1, sistema operativo utilizado, además de soportar el Visual C++ .NET, compilador utilizado.

El funcionamiento de OPC se basa principalmente en que un cliente OPC puede conectarse a servidores OPC de uno o varios vendedores. En este Proyecto la aplicación de visión artificial actúa como cliente del servidor OPC que proporciona Siemens®, fabricante del hardware de automatización empleado. Se puede construir un cliente con una interfase personalizada, para lo cual se puede usar un lenguaje de alto nivel como C++, como se ha hecho en este Proyecto. Aunque los clientes más comunes se construyen bajo una interfase automatizada que puede ser desarrollada en lenguajes como Basic o Delphi.

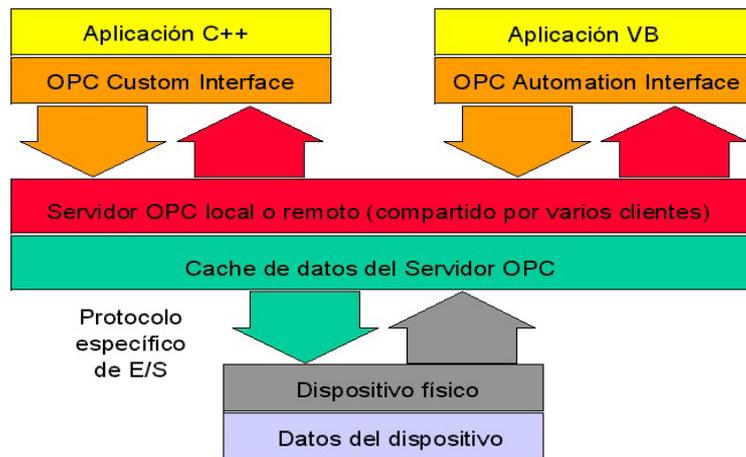


Figura 4. 31. Funcionamiento e interfaces de OPC.

4.5.3. Protocolo de Comunicación

La comunicación entre el PC y el autómatas se ha resuelto mediante un bus PROFIBUS. El autómatas utilizado dispone del hardware y software para conectarse mediante este protocolo, y para conectar el PC ya se ha explicado en el apartado anterior como se ha utilizado un módulo PROFIBUS.

PROFIBUS (**Process Field Bus**) es el más popular de los *buses de campo*. Fue desarrollado en 1989 como resultado de un proyecto de investigación en el que estaban involucradas decenas de empresas e instituciones de investigación. En Europa PROFIBUS es el sistema dominante copando más del 60% del mercado de los sistemas automatizados industriales.

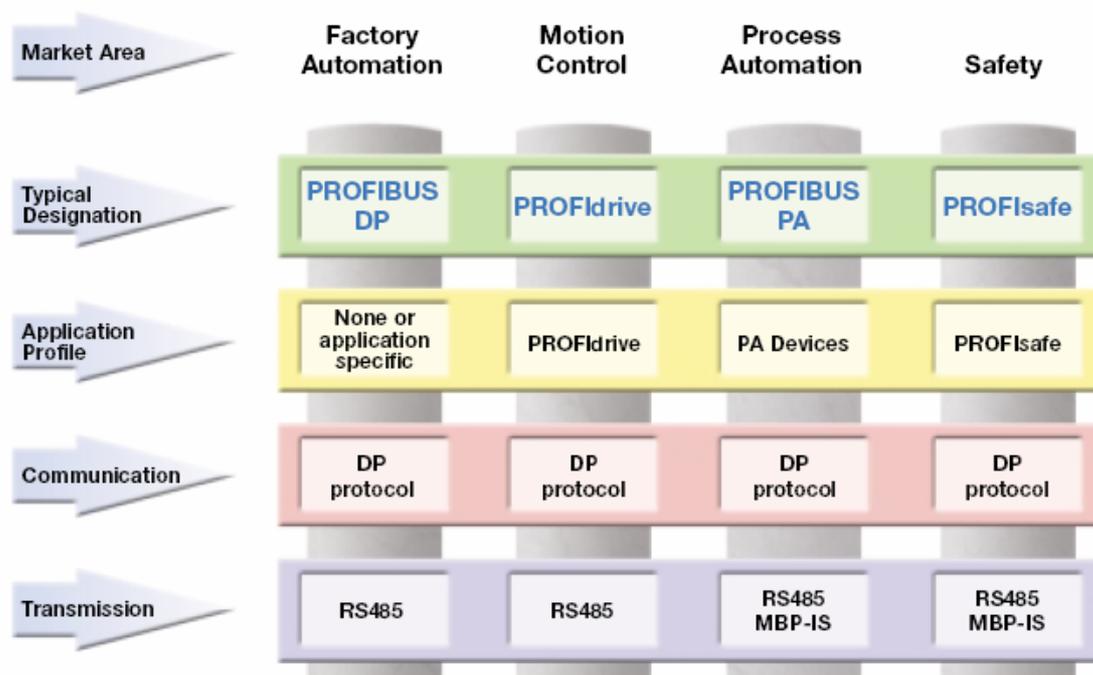


Figura 4. 32. Esquema en donde se especifica la versión de PROFIBUS a utilizar según el área.

Existen tres versiones diferentes de PROFIBUS atendiendo a su interfaz con el usuario:

- PROFIBUS-DP (Periferia Descentralizada): es utilizado para entradas y salidas remotas rápidas, opción óptima cuando se conectan sensores y actuadores a dispositivo de control.
- PROFIBUS-FMS: está basado en un modelo Cliente-Servidor y se utiliza para la comunicación de dispositivos de automatización.
- PROFIBUS-PA: se usa para conexión de dispositivos de campo y de transmisores con dispositivo de control de proceso. Permite la transmisión segura y la alimentación a través del bus.

Existen además perfiles con funcionalidad adicional para aplicaciones de seguridad, PROFIsafe, y para control de movimiento (PROFIdrive).

En el estándar PROFIBUS no solo se definen distintas versiones para el interfaz de usuario, sino que también tiene especificaciones sobre la capa física y de acceso al medio (niveles 1 y 2 del modelo de referencia OSI). Los soportes físicos también pueden ser varios, pudiéndose construir un bus PROFIBUS con:

- Fibra Óptica: para aplicaciones a larga distancia y/o donde existan muchas interferencias electromagnéticas.
- Wireless: para comunicación con componentes móviles, se suele utilizar en combinación con las otras dos opciones.
- Cable metálico: se utiliza cuando el uso de las otras dos no está justificado.

PROFIBUS se definió en 1991/1993 a través de la norma DIN 19245, que en 1996 se pasó a la EN 50170 y desde 1999 está incluido en los estándares IEC 61158/IEC 61784. El estándar PROFIBUS es mantenido, actualizado y promocionado por *PROFIBUS Internacional* [Web 6], organización sin ánimo de lucro administrada de Karlsruhe en Alemania. *PROFIBUS Internacional* es también responsable del desarrollo de PROFINET una especie de red de campo *Ethernet*.

Para este Proyecto se ha utilizado la versión PROFIBUS-DP, ya que es la que mayores velocidades de transmisión alcanza, pudiendo llegar hasta los 12Mbits/s. Esto lo convierte en ideal para arquitecturas como la usada en este Proyecto en las que un PC va a estar vigilando el estado de una autómatas que controla una serie de sensores y actuadores, el conjunto de estos sensores y actuadores forma lo que se ha llamado sistema de escaneado.

En cuanto al medio de transmisión, se ha empleado cable metálico ya que no está justificado el incremento de coste que supone la utilización de interfaces optoelectrónicas que supondría utilizar un medio de fibra óptica, ni tampoco la pérdida de robustez que implica utilizar comunicaciones vía radio. El cable metálico cumple los requisitos que exige la modesta red formada por un PC y un autómatas que se encuentran casi pegados.

Las redes eléctricas de PROFIBUS se valen de pares trenzados de cable apantallado. Para la conexión de estos cables a los dispositivos se utiliza el interfaz RS 485, el cual opera con diferencias de voltaje. Esto le supone ser menos sensible a interferencias, comunes en entornos industriales, que los interfaces que trabajan con corriente o voltaje. En la figura 4.34 se puede ver la forma de los conectores usados en el PC y el autómatas, así como el cable utilizado.

Criteria	Electrical network		Optical network			Wireless Interfacing
	RS 485 to IEC 61158/EN 50170	IEC 61158-2	Plastic	PCF	Glass	Infrared
EMC	●●●●	●●●●	●●●●	●●●●	●●●●	●●●●
Networking across buildings	●●●● ¹⁾	●●●●	●●●●	●●●● ²⁾	●●●●	—
Range	●●●● ²⁾	●●●●	●●●●	●●●●	●●●●	●●●●
Suitability for high transfer rates	●●●● ⁴⁾	—	●●●●	●●●●	●●●●	●●●●
Simple connector installation	●●●●	●●●●	●●●●	●●●● ³⁾	●●●● ³⁾	—
Simple cable routing	●●●●	●●●●	●●●●	●●●●	●●●●	—
Potential equalizing measures required	yes	yes	no	no	no	no
Performance spectrum for special applications	●●●●	●●●●	●●●●	●●●●	●●●●	—
Use with mobile terminals	●●●●	●●●●	●●●●	●●●●	●●●●	●●●●
Use in IS environment	—	●●●●	—	—	—	—

Figura 4. 33. Criterios de selección del medio de transmisión en PROFIBUS.



Figura 4. 34. Conectores PROFIBUS sobre la tarjeta CP 5611 del PC (izquierda) y sobre el autómeta (derecha), en violeta aparece el cable utilizado como medio físico.

Debido a que PROFIBUS se basa en una estructura de bus, a cada nodo que se conecta al bus hay que asignarle una dirección distinta al resto. En este Proyecto al PC se le ha asignado la dirección 3 y al autómata la dirección 2 de forma arbitraria, sin más criterio que el mencionado anteriormente de que las direcciones de cada nodo han de ser distintas y que al bus se le asigna la dirección 1 (ver figura 4.35).

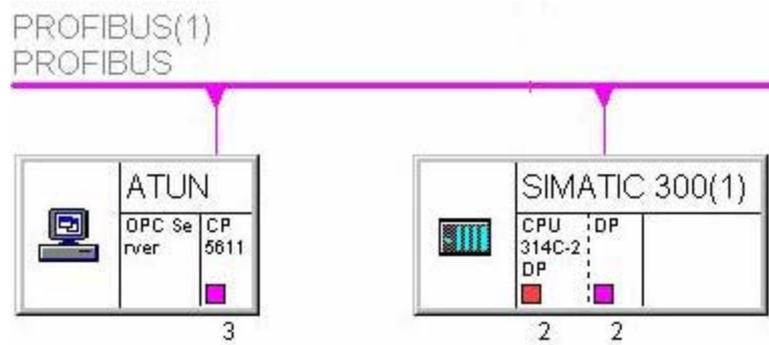


Figura 4. 35. Esquema de la red compuesta por el PC (ATUN), autómata (SIMATIC 300) y bus PROFIBUS. También se puede advertir como la comunicación de la aplicación de visión artificial con el módulo PROFIBUS se hace a través de servidor OPC.

4.6. El Autómata

Las decisiones de arranque y paro se hacen desde la aplicación de visión artificial residente en el PC. Sin embargo, el control de los sensores y actuadores que componen el sistema de escaneado corre a cargo del autómata, limitándose la aplicación de visión artificial a visualizar el estado del sistema de escaneado a través del autómata para saber si debe o no capturar imágenes, además de ir indicándole al autómata cuando debe de cambiar de estado.

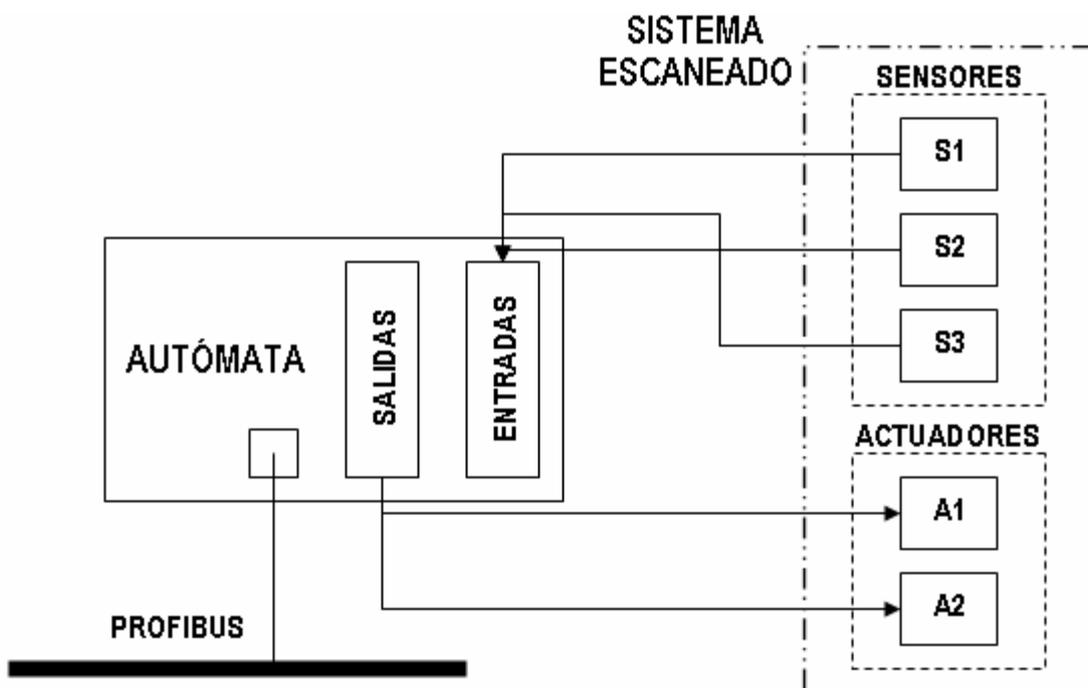


Figura 4. 36. Esquema con las conexiones del autómata.

Se ha utilizado un autómata de la compañía Siemens® perteneciente a la familia S7 300, que se caracteriza por estar formada por PLCs compactos, rápidos, de alto rendimiento y versátiles. Características que han facilitado la realización del sistema de control del sistema de escaneado.

4.6.1. La CPU del Autómata

La CPU que porta el autómata es la denominada CPU 314C-2 DP, la cual a grandes rasgos dispone de:

- De una tarjeta de memoria *Micro Memory Card SIMATIC*, imprescindible para el funcionamiento de la CPU.
- Un interfaz MPI de 9 polos.
- Un interfaz DP de 9 polos.
- 24 entradas y 16 salidas digitales.
- 4+1 entradas y 2 salidas analógicas integradas.
- 4 contadores y 1 canal de posicionamiento.

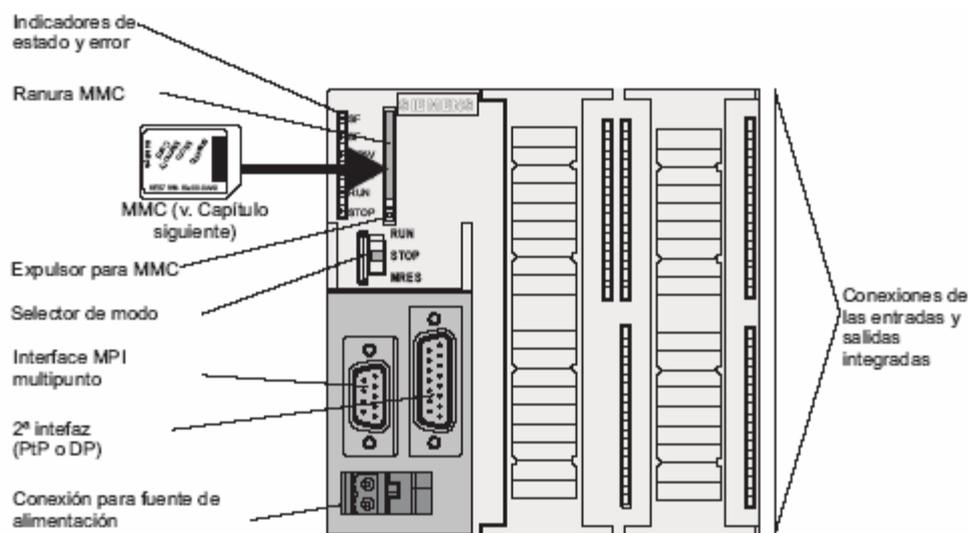


Figura 4. 37. Elementos y estructura de la CPU 314C-2 DP.

En la figura 4.37 se puede apreciar como aparece un selector de modo de operación, que como su nombre indica sirve para poner al autómata en tres modos de operación diferentes (figura 4.38). También dispone de una serie de indicadores LED para poder explorar el estado en que se encuentra la CPU, según se resume en la figura 4.38. Además de las partes visibles en este esquema, también posee una fuente de alimentación conectada modelo PS 307 2A fabricada por Siemens®.

Como módulo de memoria para esta CPU se utiliza un *Micro Memory Card SIMATIC* (MMC), éste se puede utilizar como memoria de carga o como soporte transportable. Es imprescindible para el funcionamiento de la CPU. La MMC se ha utilizado en este Proyecto para guardar los datos sobre el programa de usuario (se encargará del control del sistema de escaneado), y de los datos de configuración sobre la red PROFIBUS y la CPU (definidos en forma de proyecto del Administrador de SIMATIC ó STEP 7 ver apartado 7.2).

Entonces la MMC actúa como memoria de carga y es extraíble, no obstante la CPU posee una memoria interna dividida en *memoria de trabajo* que sirve para procesar el código y los datos del programa de usuario, y *memoria del sistema* con las áreas de operandos (marcas, temporizadores y contadores), las imágenes del proceso de entradas y salidas, y los datos locales.

La CPU dispone de dos interfaces, el primero de ellos es el MPI (*Multi Point Interface*) y que trabaja con RS 485. La velocidad de transferencia que utiliza es de 187.5 Kbaudios, no permitiendo cambiarla. La CPU envía sus parámetros a través interfaz MPI automáticamente (por ejemplo la velocidad de transferencia), de este modo, se pudo conectar por primera vez al PC con el autómat. De manera que este interfaz sirvió para que en un primer momento, cuando aún la red PROFIBUS no estaba creada, enviar la configuración al autómat para crear el enlace PROFIBUS hecho desde PC utilizando el software STEP 7 (ver apartado 7.2).

Posición	Significado	Explicaciones
RUN	Modo RUN	La CPU procesa el programa de usuario.
STOP	Modo STOP	La CPU no procesa ningún programa de usuario.
MRES	Borrado total	Posición no enclavable del selector de modo para el borrado total de la CPU. El borrado total por medio del selector de modo de operación requiere una secuencia especial de operación (consulte el manual de instalación, capítulo <i>Puesta en servicio</i>).

Indicadores de la CPU:

- SF (rojo) Fallo de hardware o software
- BF (rojo) Fallo de bus (sólo CPU 313C-2 DP y 314C-2 DP)
- DC5V (verde) Alimentación de 5 V para CPU y bus S7-300 funciona correctamente.
- FRCE (amar.) Petición de forzado activa.
- RUN (verde) CPU en RUN; LED parpadea en el arranque a 2 Hz; en PARADA a 0,5 Hz.
- STOP (amar.) CPU en STOP o en PARADA o en arranque; El LED parpadea a 0,5 Hz al solicitar borrado, durante el borrado a 2 Hz.

Figura 4. 38. Posiciones del selector de modo (arriba) y display de estado y error de la CPU 314C-2 DP (abajo).

También dispone de un interfaz PROFIBUS DP que ha servido principalmente para la conexión de periferia descentralizada, DP, entre el PC y el autómat. Este interfaz, como ya se ha comentado, se configuró desde el PC mediante STEP 7 y cuyos parámetros se enviaron a la CPU vía MPI en donde fueron almacenados en la tarjeta de memoria MMC. PROFIBUS-DP puede alcanzar unas velocidades de transferencia de 12Mbaudios.

4.6.2. Entradas y Salidas del Autómat

La CPU utilizada dispone de dos módulos de entrada y salida, uno para digitales y otro para las analógicas. Las entradas digitales se han conectado a los interruptores de proximidad usados en el sistema de escaneado, de forma que la entrada correspondiente se activará con la llegada de un flanco ascendente de tensión. Las características de estas entradas se resumen en la tabla siguiente, y se han tenido muy en cuenta al realizar el conexionado entre los sensores y el autómat:

Datos Técnicos	
Datos específicos del módulo CPU 314C-2	
Cantidad de entradas	24
Longitud del cable	
• sin apantallar	Máx. 600m
• apantallado	Máx. 1000m
Tensión, intensidades, potenciales	
Tensión nominal de carga L+	24Vcc
• Protección contra inversiones de polaridad	Sí
Cantidad de entradas accesibles simultáneamente	
• Hasta 40°C	24
• Hasta 60°C	12
Separación galvánica	
• Entre canales y bus posterior	Sí
• Entre los canales	No
Diferencia de potencial admisible entre circuitos diferentes	75Vcc/60Vca
Aislamiento ensayado con	500Vcc
Consumo de la tensión de carga L+ (sin carga)	Máx. 70mA
Estado y diagnóstico	
Indicación de estado	Un LED verde por canal

Tabla 4. 2. Datos técnicos de las entradas digitales de la CPU 314C-2 DP.

Dos de las salidas digitales del autómatas se conectan al puerto de entrada de la caja de electroválvulas, para así controlar el estado de las dos válvulas que controlan las entradas de aire del actuador neumático (ver apartado 4.7.2). Para la realización de estas conexiones se han de tener en cuenta los datos técnicos de estas salidas:

Datos Técnicos	
Datos específicos del módulo CPU 314C-2	
Cantidad de entradas	16
• de ellas, salidas rápidas	4
Longitud del cable	
• sin apantallar	Máx. 600m
• apantallado	Máx. 1000m
Tensión, intensidades, potenciales	
Tensión nominal de carga L+	24Vcc
• Protección contra inversiones de polaridad	Sí
Total corriente de las salidas (por grupo)	
• Hasta 40°C	Máx. 3.0A
• Hasta 60°C	Máx. 2.0A
Separación galvánica	
• Entre canales y bus posterior	Sí
• Entre los canales	Sí
Diferencia de potencial admisible entre circuitos diferentes	75Vcc/60Vca
Aislamiento ensayado con	500Vcc

Consumo de la tensión de carga L+	Máx. 100mA
Estado y diagnóstico	
Indicación de estado	Un LED verde por canal

Tabla 4. 3. Datos técnicos de las salidas digitales de la CPU 314C-2 DP.

Para el caso de la lectura de la señal que envía sensor de ultrasonidos utilizado por el sistema de escaneado, debido a su carácter analógico, se realiza a través de las entradas analógicas. Luego para la conexión del sensor de ultrasonidos a una entrada analógica (ver apartado 4.7.4), es necesario tener en cuenta los datos técnicos del módulo de entradas analógicas. En concreto las que trabajan con intensidad, ya que el sensor de ultrasonidos trabaja enviando señales codificadas en intensidad de corriente y no en diferencias de potencial.

Datos Técnicos	
Datos específicos del módulo CPU 314C-2	
Cantidad de entradas	4 canales de entrada de tensión e intensidad 1 canal de entrada de resistencia
Longitud del cable	
• apantallado	Máx. 100m
Tensión, intensidades, potenciales	
Separación galvánica	
• Entre canales y bus posterior	Sí
• Entre los canales	No
Diferencia de potencial admisible	
• Entre las entradas y M_{ANA} (U_{CM})	1.0Vcc
• Entre M_{ANA} y M_{Intern} (U_{ISO})	75Vcc/60Vca
Aislamiento ensayado con	600Vcc
Formación de valores analógicos	
Principio de medida	Codificación momentánea (aproximaciones sucesivas)
Tiempo de integración/conversión/resolución (por canal)	
• Parametrizable	Sí
• Tiempo de integración en ms	2.5/16.6/20
• Frecuencia de entrada admisible	Máx. 400Hz
• Resolución (incl. rebase de margen)	11 bits + signo
• Supresión de perturbaciones de tensión para frecuencia perturbadora f_1	400/60/50Hz
Constante del filtro de entrada	0.38ms
Tiempo de ejecución básico	1ms
Límites de error	
Límite de error práctico (en todo el margen de temp., referido al margen de entrada)	< 1%
Límite de error básico (límite de error práctico a 25°C, referido al margen de entrada)	< 0.7%
Error por temperatura (referido al margen de entrada)	$\pm 0.006\%/K$
Error por linealidad (referido al margen de entrada)	$\pm 0.06\%$

Exactitud de repetición (en estado estacionario a 25°C, referido al margen de entrada)	± 0.06 %
Datos para seleccionar un sensor	
Márgenes de entrada (valores nominales)/resistencia de entrada	4mA a 20mA/50Ω
Tensión de entrada admisible (límite de destrucción)	Máx. 2.5V duradero
Intensidad de entrada admisible (límite de destrucción)	Máx. 50mA duradero
Linealización de característica	Con software
Compensación de temperatura	No

Tabla 4. 4. Datos técnicos de interés de las entradas analógicas por intensidad de la CPU 314C-2 DP.

Además la CPU 314C-2 DP también posee un módulo de salidas analógicas, no obstante, puesto que no se utiliza en este Proyecto no es necesario escribir en esta memoria sus características. Nótese que en las tablas anteriores sólo se han especificado las características interesantes para este Proyecto de los tipos de entrada/salida utilizado, si se quiere más información sobre esto ver los manuales existentes en [Web 5].

La CPU junto con su fuente de alimentación y módulos de entrada/salida está montada sobre un soporte que dispone de un tablero con conectores banana hembra, en donde tiene mapeadas las entradas y salidas. De esta forma se pueden conectar los sensores y actuadores al autómata de forma rápida a través de conectores banana. Así se ha hecho en este Proyecto, además de utilizarse los interruptores que dispone el tablero como entradas digitales para la introducción manual de datos, algo común durante la fase de desarrollo del programa de control del autómata y útil en la ejecución en modo manual de la aplicación de visión (ver apartado 7.4.1). También desde el tablero se puede acceder a la fuente de alimentación, para así poder alimentar a los sensores que necesiten alimentación externa, como es el caso del sensor de ultrasonidos.

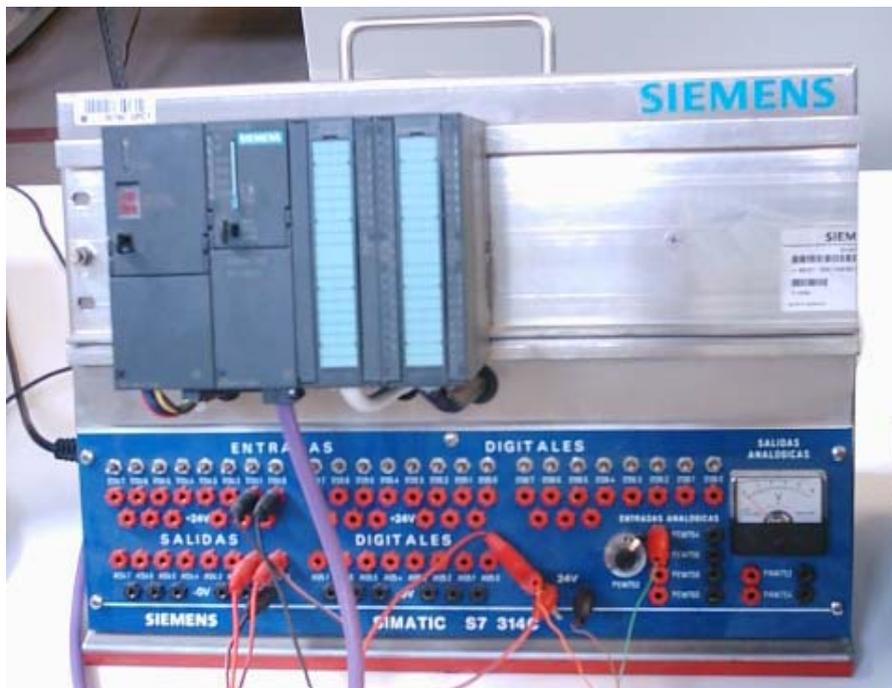


Figura 4. 39. Soporte del autómata con tablero eléctrico que permite la conexión con las entradas y salidas usando conectores tipo banana.

4.7. Sistema de Escaneado

Para el correcto funcionamiento de la aplicación de visión artificial se hace necesaria una correcta iluminación de la nube pulverizada, para ello es necesario hacer cortes con la luz proveniente del sistema de iluminación sobre esta nube. Ya se ha comentado en el apartado 4.2.1 como la fuente de iluminación posee un diagrama de radiación que se puede aproximar a un plano. Luego el sistema de escaneado, a partir de las señales de control que le envía el autómatas, se limita a mover la fuente óptica en sentido vertical (estando el plano definido por el haz de luz paralelo al suelo del cajón del túnel), realizando paradas en cada corte (a diferentes altitudes) para dar tiempo a la aplicación a tomar las imágenes que necesite. Al finalizar el sistema de escaneado volverá a su posición original, esto es, el corte más cercano al suelo.

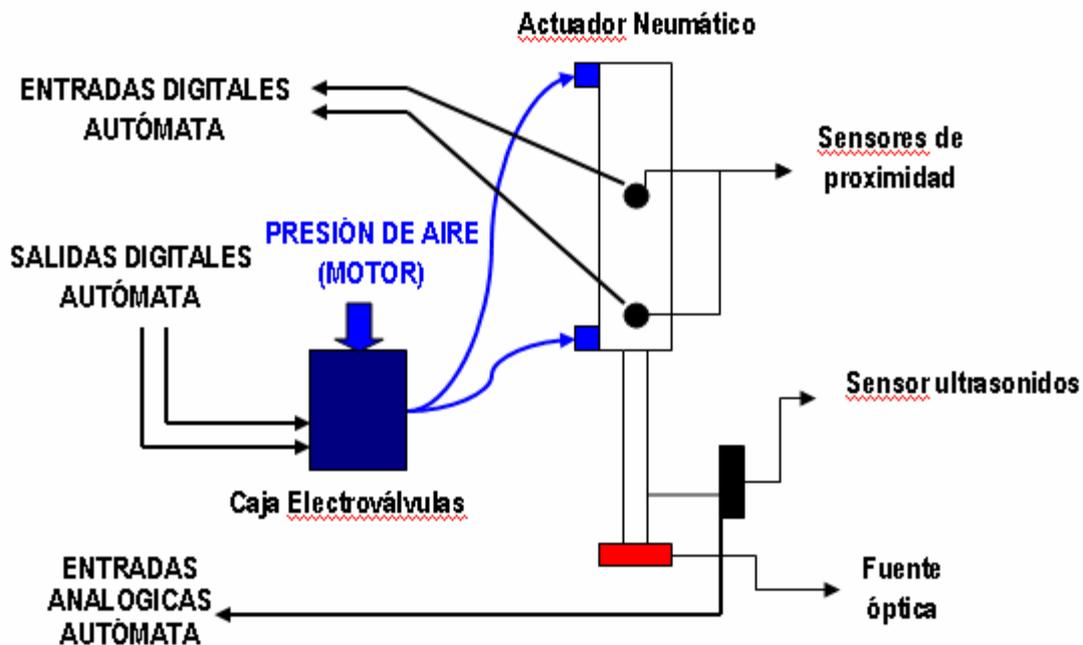


Figura 4. 40. Esquema del sistema de escaneado, las flechas negras en trazo grueso representan señales eléctricas y las azules señales de aire comprimido.

El sistema de escaneado está compuesto por un actuador neumático que es el encargado de realizar los movimientos, una caja de electroválvulas que se encarga de traducir las señales eléctricas de control provenientes del autómatas en señales de presión de aire, dos sensores de proximidad que se encargan de definir los extremos máximos que debe alcanzar el actuador, y un sensor de ultrasonidos que mide la distancia al suelo del sistema de iluminación permitiendo al autómatas decidir cuando se ha llegado a una zona donde parar por un tiempo (corte).

El uso de un sistema neumático, que quizás no sea el más adecuado para realizar este sistema, viene justificado por que se aprovechan las piezas compradas para el prototipo AUTOMAN. De esta forma se reducen costes, así como se acumula experiencia que posteriormente facilitará el diseño y construcción del sistema de control.

4.7.1. Actuador Neumático

La base del sistema de escaneado la constituye el actuador neumático debido a que es encargado de mover la fuente de iluminación, la cual se encuentra adherida al extremo externo

del vástago, lugar donde también se fijará el sensor de ultrasonidos. Así el movimiento del vástago debido a la entrada de aire a presión por la conexión neumática que corresponda, conllevará también un movimiento de la fuente óptica y del sensor de ultrasonidos. El actuador neumático usado en este Proyecto pertenece al fabricante Festo® y se corresponde con un cilindro normalizado DNC según la norma ISO 15552.



Figura 4. 41. Fotografía del actuador neumático y del montaje de la fuente óptica y sensor de ultrasonidos sobre el extremo de su vástago. También se aprecian las guideras de madera e hierro (en negro).

El actuador posee dos entradas neumáticas una en cada uno de sus extremos, de modo que al entrarle aire comprimido en la entrada del extremo superior (ver parte izquierda de la figura 4.42), desplazará al vástago, y por tanto a la fuente de iluminación y al sensor de ultrasonidos, hacia abajo. Si el aire comprimido entra por la otra entrada neumática el vástago, con la fuente de iluminación y sensor de ultrasonidos, se moverá hacia arriba. Entonces, conectando estas entradas neumáticas a la caja de electroválvulas se puede controlar el movimiento del actuador desde el autómatá.



Figura 4. 42. Estructura interna del actuador neumático (izquierda). Entrada neumática superior con su válvula reguladora de caudal y racor conectado un tubo flexible (azul) proveniente de la caja de electroválvulas.

Sobre cada una de las entradas neumáticas se ha colocado una válvula reguladora de caudal para poder controlar la velocidad del actuador, ya que la señal proveniente de las electroválvulas es “abierto o cerrado”, esto es, existencia o ausencia de aire comprimido. En este Proyecto se han cerrado un poco estas válvulas, desde el tornillo que se provee como interfaz (ver derecha imagen 4.42), de manera que el actuador tenga movimientos más lentos, pero suficientemente precisos. A las entradas neumáticas se ha conectado también un racor de roscado rápido, para poder conectar los tubos flexibles que han facilitado la conexión con la caja de electroválvulas.

El actuador o cilindro neumático utilizado en este Proyecto tiene un problema, y es que su vástago tiene una sección circular, lo que implica que gira en un sentido u otro a la vez que se mueve hacia arriba y hacia abajo. Esto supondría un giro de la fuente de iluminación, que expodría al sistema de visión artificial a un funcionamiento incorrecto. Además, también giraría alrededor del vástago el sensor de ultrasonidos, lo que puede desembocar en un funcionamiento incorrecto de este sensor, o incluso en su rotura debido a un choque con la tijera que sujeta al pulverizador. Por esto se han hecho necesarias dos guideras, una para el sensor de ultrasonidos de madera, y otra para la fuente de iluminación de hierro recubierto con cinta aislante negra para facilitar el deslizamiento (ver figura 4.41), evitándose así el giro de estos dispositivos cuando el vástago se mueve. La utilización de este actuador se debe a la adecuación de su tamaño, 350 mm de carrera, a la distancia de escaneado y a que se encontraba disponible en el laboratorio donde se va ha realizado este Proyecto. Lo ideal habría sido utilizar un actuador con un vástago de sección cuadrada, pudiéndose entonces prescindir de las guideras.

4.7.2. Terminal de Válvulas

En este Proyecto para transformar las señales eléctricas de control enviadas por el autómeta, por señales de aire comprimido que alimenten el actuador neumático, se ha utilizado un caja de electroválvulas o Terminal de Válvulas de Festo® [Web 7], en concreto el modelo 10 CPV de 8 parejas de válvulas. Se trata de un terminal ligero y compacto que permite un montaje sencillo. El uso de este terminal se debe a que se ha querido aprovechar el terminal que el prototipo AUTOMAN utilizará, compartiendo una de sus ocho parejas de válvulas con este Proyecto (AUTOMAN-TUNEL).

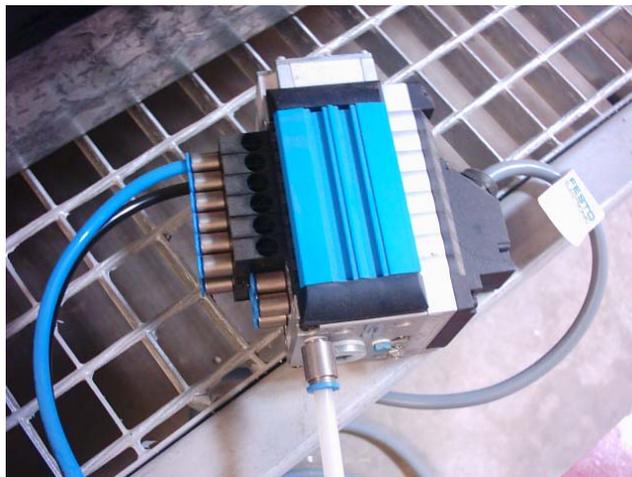


Figura 4. 43. Caja de electroválvulas, en el tubo blanco le entra el aire comprimido del motor, del cable gris le llegan las señales de control del autómeta, y de los tubos negro (va a la entrada inferior) y azul (entrada superior) sale el aire comprimido para el actuador desde la pareja de válvulas 1.

Al terminal de válvulas le llega el aire comprimido sin lubricar, como lo exigen los datos técnicos [Web 7], para su alimentación desde un motor eléctrico a través de un tubo flexible blanco de 8mm de diámetro (ver figuras 4.43 y 4.44). Para controlar la entrada inferior del actuador, encargada de subir el vástago, se utilizará la válvula conectada al tubo negro, y para la entrada superior, que baja el vástago, la conectada al tubo azul. El ancho de los conectores de las válvulas de 6mm y para todas las conexiones se han utilizado conectores rápidos.



Figura 4. 44. Conexión entre motor y terminal de válvulas mediante un tubo blanco, para la alimentación de aire comprimido.

Las señales de control llegan al terminal desde el autómata a través de un conector multipolo (ver figura 4.43) permitiendo integrar la parte neumática y la parte eléctrica, y estableciendo una conexión con el terminal mediante un solo conducto. El conector es de tipo Sub-D, posee 25 contactos, y el cable es un IP65 de cloruro de polivinilo con 25 hilos en su interior. Las correspondencias entre válvulas, conectores e hilos del cable se resumen en la figura 4.45, según esta, para este Proyecto solo utilizará el hilo blanco (controla la válvula conectada al tubo negro) que irá conectado a la salida digital 1 del autómata, y el verde (controla la válvula conectada al tubo azul) conectado a la salida digital 0 del autómata. Además del hilo negro que se conecta la masa del autómata. Las conexiones eléctricas del conjunto del sistema de escaneado con el autómata se explican de una forma más gráfica en el apartado 4.7.5.

La parte eléctrica de las válvulas es activada con 24V, activándose con reducción de corriente para ahorrar energía y disminuir calor. Al desconectar los picos tensión se limitan a 38V. Estas características las convierten en compatibles con las salidas digitales (apartado 4.6.2) del autómata permitiendo una conexión directa con éste. Las velocidades de conexión y desconexión de las válvulas son de 17 y 25ms respectivamente. El terminal dispone de un

conjunto de LEDs que permiten visualizar el estado de cada válvula, esto es muy útil para la detección de fallos en el sistema.

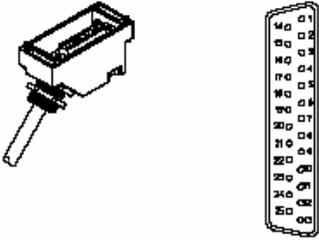
Vista del conector	Pin	Color del hilo	Válvula 24 V DC	
Cable con conector Sub-D de 25 polos para terminales de válvulas séxtuplos u óctuplos				
	1	Bianco	1	14
	2	Verde		12
	3	Amarillo	2	14
	4	Gris		12
	5	Rosa	3	14
	6	Azul		12
	7	Rojo	4	14
	8	Violeta		12
	9	Gris y rosa	5	14
	10	Rojo y azul		12
	11	Bianco y verde	6	14
	12	Marrón y verde		12
	13	Bianco y amarillo	7	14
	14	Amarillo y marrón		12
	15	Bianco y gris	8	14
	16	Gris y marrón		12
	17			
	18			
	19			
	20			
	21			
	22			
	23			
	24	Marrón	(0 V) ²	
	25	Negro	(0 V) ²	

Figura 4. 45. Ocupación de los contactos para el cable multipolo utilizado (plano en el sentido de conexión).

4.7.3. Sensores de Proximidad

Para conocer cuando el actuador ha alcanzado los niveles superior e inferior a los que debe de llegar la fuente de iluminación, se utilizan dos sensores de proximidad. También sirve para evitar que el vástago del actuador se mueva fuera de los límites, tanto inferior como superior, permitidos, evitando así la colisión de la fuente óptica y/o sensor de ultrasonidos con la tijera que sujeta el pulverizador. Además, las posiciones de la fuente de iluminación que estos sensores determinan serán utilizadas como dos de los cortes que define el sistema de escaneado.

Sin embargo, toda la funcionalidad proporcionada por estos sensores puede ser sustituida aprovechando la información que genera el sensor de ultrasonidos. No obstante, se ha considerado utilizar estos sensores ya que confieren una mayor robustez al sistema aumentando la confiabilidad en la detección de movimientos extremos. El uso de solo el sensor de ultrasonidos puede implicar situaciones de peligro ya que no es tan rápido, y en ocasiones pierde el eco dejando de funcionar momentáneamente. En un momento dado, un fallo en la simulación por culpa del sensor de ultrasonidos es admisible, pero lo que no es admisible es una rotura en algún dispositivo del sistema, es por esto que se usan los sensores de proximidad.

También se podría pensar en el uso de varios sensores de proximidad en lugar del sensor de ultrasonidos para la determinación de las posiciones de corte. Sin embargo, es una solución poco escalable en el sentido si incrementa el número de cortes, rápidamente se acaba el espacio para su colocación sobre el actuador neumático, así como el número de entradas digitales del autómatas.

El tipo de sensor usado en este Proyecto es detector de proximidad SME-8F de Festo® [Web 7], el mismo fabricante que el cilindro neumático. Estos sensores han sido diseñados para la detección del émbolo en el tipo de cilindro como el utilizado en este Proyecto, que poseen ranura para detectores integrada. Los sensores disponen de un sistema de fijación para sujetarlos fuertemente una vez dentro de la ranura del cilindro.

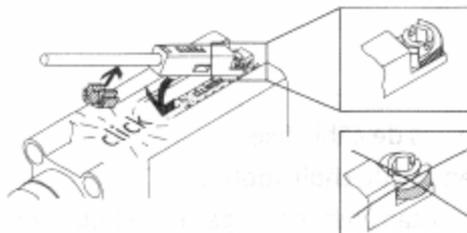


Figura 4. 46. Sistema de fijación del sensor al cilindro neumático.

El sensor dispone de un conector de 3 pines, cada uno identificado por un número y conectados a hilos de color distinto. Al pin 1 se le conecta el hilo marrón, que ha su vez es conectado a la salida de alimentación de 24Vcc del tablero eléctrico del autómeta. A la salida del pin 4 (hilo negro) se conecta una resistencia de carga de $1k\Omega$ que por el otro extremo, junto con el pin 3 (hilo azul), se conecta a la entrada digital del autómeta. Según este conexionado cuando el émbolo está cerca activa al sensor magnético que cierra un interruptor que separa los pines 1 y 4, entonces los 24Vcc pasan al pin 4, y por tanto a la entrada digital, capturando el autómeta el flanco ascendente de tensión. El sensor colocado en la parte superior del cilindro neumático se conectará con la entrada digital 0 del autómeta y el inferior con la 1. Según el conexionado explicado anteriormente.

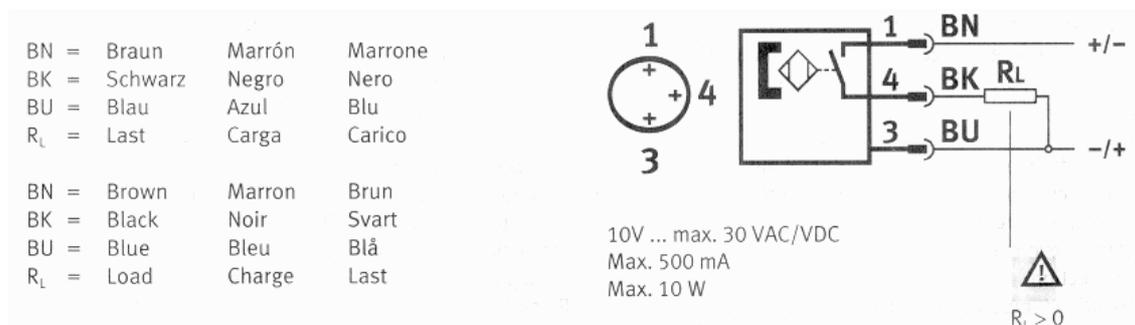


Figura 4. 47. Esquema eléctrico y conector de un sensor de proximidad.

El sensor trabaja con voltajes de entre 10 y 30V de corriente continua y alterna, lo que permite alimentarlo con los 24Vcc que provee la fuente de alimentación del autómeta, y conectarlo a sus entradas digitales de 24Vcc. La inclusión de la resistencia de carga evita que se alcancen intensidades sobre el sensor de 500mA que lo destruirían, en concreto la corriente será de aproximadamente $24V/1000\Omega = 24mA$. Para el cálculo exacto se necesitaría conocer la resistencia interna del interruptor y de la entrada digital del autómeta, pero en cualquier caso será menor de estos 24mA. Al introducir la resistencia de 1000Ω también se cumple la restricción de carga máxima de 10W, al ser esta de aproximadamente $24V \cdot 0.024A = 0.575W$.

Estos sensores trabajan con un tiempo de conmutación menor de 0.5ms lo que sumando a su precisión (repetibilidad $\pm 0.1mm$), y rapidez del autómeta en la detección de flancos (3ms), se

pueda generar una respuesta rápida y precisa a su activación. Los cortes de la nube generados por los sensores de proximidad son muy precisos, en el sentido de que el error en su altura es muy pequeño en comparación con los generados por el sensor de ultrasonidos. Para calcular el tiempo de respuesta total habría que añadirle el tiempo de desconexión de la válvula, 25ms, y el tiempo del software del autómatas en identificar la señal y enviar un cero sobre el conector de la válvula correspondiente, sobre este último tiempo sólo se podría hacer una estimación de tiempo máximo. En cualquier caso la precisión en las paradas del sistema de escaneado que se alcanza con estos sensores es más que suficiente.

4.7.4. Sensor de Ultrasonidos

Para determinar a que altura se encuentra la fuente de iluminación, y por tanto decidir donde se hacen las paradas o cortes sobre la nube, se utiliza un sensor de ultrasonidos fabricado por Honeywell International Inc [Web 8], más concretamente es el modelo 943-F4Y-2D-1D0-360E. Para medir la distancia del sistema de iluminación también se encuentra cogido sobre el vástago del actuador, con la cara que envía los ultrasonidos y recibe los ecos, *sensing face*, paralela al suelo del cajón (para recoger correctamente los ecos), superficie sobre la que se quiere medir la distancia. El sensor genera una señal analógica en forma de corriente que es recogida por el autómatas por una entrada analógica de corriente.

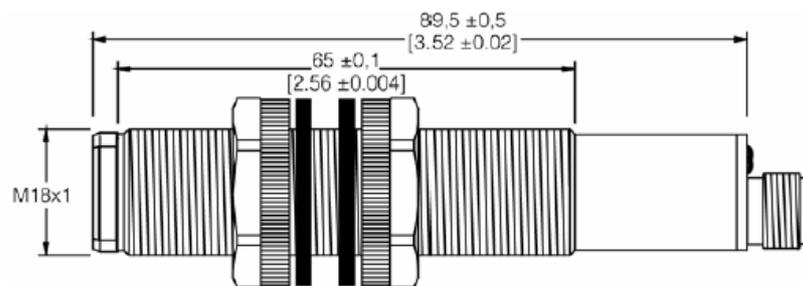


Figura 4. 48. Aspecto y dimensiones del sensor de ultrasonidos utilizado. En el extremo derecho está el conector y en el izquierdo se encuentra la *sensing face*.

La sujeción de este sensor sobre el vástago en este proyecto ha sido condicionada por dos aspectos. En primer lugar, debido a que la zona muerta del sensor es de 30mm. La zona muerta en un sensor de ultrasonidos es la mínima distancia a la que puede estar el objetivo para poder ser medido. Si el objetivo está muy cerca (30mm en este caso), el flanco inicial del pulso de sonido enviado por el sensor estará siendo recibido cuando aún continúa transmitiendo ese mismo pulso, siendo esta información rechazada por el sensor. El sensor está diseñado para trabajar con máxima precisión hasta los 400mm, ajustándose perfectamente a las necesidades de este Proyecto.

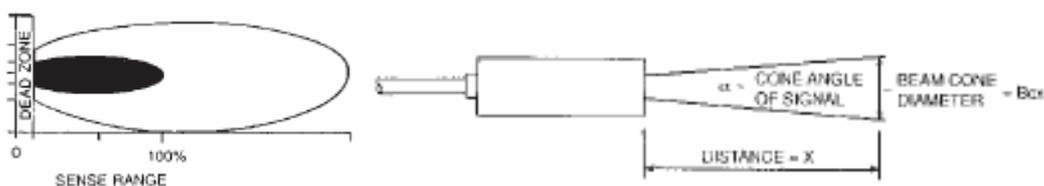


Figura 4. 49. Conceptos de zona muerta, ancho de haz y diámetro de detección en un sensor de ultrasonidos.

El segundo factor, es el ancho de haz, definido como el ángulo del cono en que la intensidad del pulso decae 3dB. Fuera de este ángulo los objetivos pueden no ser detectados, en este Proyecto se quiere detectar suelo luego no es este el problema. El problema es que si el sensor se sitúa junto a la fuente de iluminación la pared del cajón caerá dentro del cono de detección pudiendo alterar las medidas. Por esto, hay que separar al sensor del extremo del vástago que está muy cerca de la pared. Como el ancho de haz del este sensor es de 8° , y en la situación más desfavorable, que se corresponde cuando el sensor se encuentra arriba lo más arriba permitido que se corresponde con unos 40cm, el sensor a de estar separado:

$$D_{3dB} = 2 \cdot d \cdot \tan\left(\frac{\alpha_{3dB}}{2}\right) = 2 \cdot 40\text{cm} \cdot \tan\left(\frac{8^\circ}{2}\right) \approx 5.6\text{cm} \quad (4.7) \text{ [Web 8]}$$

Es por estos dos motivos expuestos en los párrafos anteriores, que el sensor se ha colocado sobre vástago encima de la fuente de iluminación y separado de la pared una distancia superior a 5.6cm, gracias un sistema de sujeción de la figura 4.50.



Figura 4. 50. El sensor de ultrasonidos (en azul) se encuentra separado del vástago gracias a un soporte con una barra metálica.

El sensor genera una señal en intensidad de corriente que varía linealmente con la distancia al objetivo (suelo del cajón). Esta señal varía entre 4 y 20mA entre la distancia máxima y mínima definida para el sensor, en este Proyecto se ha definido como distancia menor 19cm y como mayor 37cm, rango por el que el sensor se moverá movido por el vástago. Al ajustar el rango del sensor lo que se consigue es maximizar la sensibilidad. Además se le saca todo el partido posible a la sensibilidad de este sensor porque el suelo del cajón es metálico, por lo que refleja muy bien los pulsos de ultrasonidos.

Este sensor dispone de un conector con 5 pines que se corresponden con el color del hilo al que se encuentran conectados. El cable formado por estos hilos es menor de 2m como se especifica en los datos técnicos del sensor [Web 8], lo que es suficiente para llegar hasta el automático. El pin 1 (hilo marrón) se conecta a la alimentación del tablero del automático de 24Vcc. El pin 4 (hilo negro) se conecta la entrada analógica de corriente 0 del automático, y es por donde se envía la información en forma de intensidad de corriente. El pin 3 (hilo azul) es el conector de tierra que se conectará a la del automático. En el pin 5 (hilo rosa) se conecta la entrada de programación (*Teach-In*), de manera que los comandos de programación se generan cortocircuitando este pin con el de tierra (ver apartado 7.3.3 en donde se explica como calibrar este sensor).

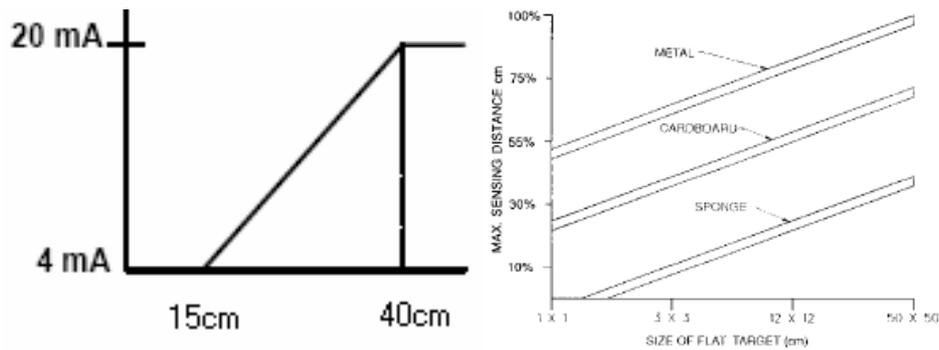


Figura 4. 51. Función de transferencia del sensor de ultrasonidos según la configuración de este Proyecto (izquierda), derecha sensibilidad del sensor en función del material del objetivo (derecha).

En general, este sensor posee unas prestaciones peores que los de proximidad, debido a una mayor lentitud de respuesta del sensor, 60ms frente a 0.5ms, además de que el autómatas ha digitalizar la señal analógica que éste envía, lo que supone otro retardo extra. También tiene una precisión intrínseca (repetibilidad $\pm 1\text{mm}$) y fiabilidad menores. Todo esto redundará en una menor precisión, en cuanto a las paradas del sistema de escaneado, realizadas con información del sensor de ultrasonidos con respecto a las realizadas con los sensores de proximidad. No obstante, la precisión es suficiente y se justifica su uso en aras de permitir un fácil escalado del sistema, en el sentido de aumentar el número de paradas como se explica en el apartado anterior. Además, permite acumular experiencia valiosa en su uso, útil para cuando se vaya a implementar el prototipo AUTOMAN.

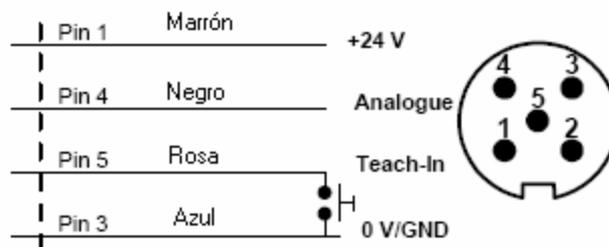


Figura 4. 52. Correspondencia entre pin, color de hilo y función (izquierda). Dibujo del conector visto desde arriba (derecha).

4.7.5. Conexiones con el Autómata

En el apartado correspondiente a cada sensor o actuador se ha comprobado el cumplimiento de los requisitos eléctricos según lo requería el sensor o actuador para su correcto funcionamiento. Ahora falta comprobar como el conjunto de conexiones respetan los requisitos impuestos por el autómatas (las tablas de datos técnicos de entradas y salidas del autómatas están en el apartado 4.6.2).

La fuente de alimentación del autómatas es capaz de proporcionar 2A y va a ser la encargada de alimentar los sensores de proximidad y de ultrasonidos. Los sensores de proximidad normalmente se encuentran desactivados por lo que solo tendrá que alimentar al sensor de ultrasonidos que consume como máximo 20mA. En el peor de los casos sucede cuando hay que alimentar también a un sensor de proximidad, ya que por construcción del sistema (y sino se considera una avería) los dos sensores de proximidad no pueden estar conectados a la vez, el

consumo total de la fuente es de $24+20 = 44\text{mA}$ ($44\text{mA}\cdot 24\text{V} = 1.026\text{W}$), que es bastante menor de los 2A que puede proporcionar.

Las salidas digitales pueden proporcionar un máximo de 100mA suficiente para alimentar cada una de las electroválvulas que consumen 25mA ($25\text{mA}\cdot 24\text{V} = 0.6\text{ W}$). Las entradas digitales soportan cada una corriente máxima de 70mA la cual es menor de 24mA, que es la corriente que les entra proveniente de los sensores de proximidad. Y la entrada analógica por corriente tiene unos márgenes de entrada de entre 4 y 20mA, ajustándose perfectamente a los márgenes de corriente que proporciona el sensor de ultrasonidos. Este último aspecto además condiciona una óptima digitalización de la señal analógica proveniente del sensor de ultrasonidos.

A modo de resumen en los siguientes esquemas se dibujan las conexiones de los sensores y actuadores con el tablero eléctrico del autómatas. Es importante recordar, que las entradas y salidas digitales, así como entradas analógicas de corriente, del tablero están mapeadas con las del autómatas.

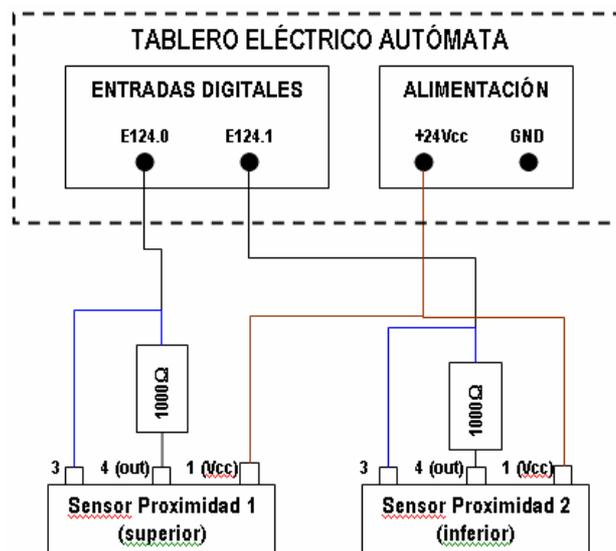


Figura 4. 53. Conexiones eléctricas entre sensores de proximidad y tablero eléctrico del autómatas.

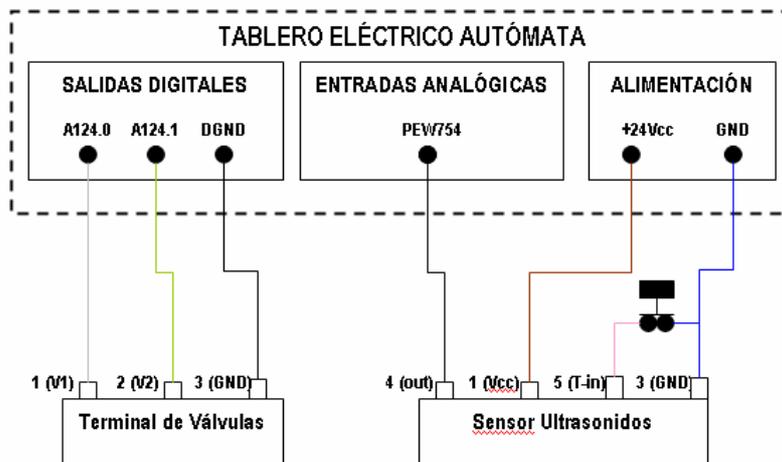


Figura 4. 54. Conexiones eléctricas entre el tablero eléctrico del autómatas con el sensor de ultrasonidos y terminal de válvulas.

CAPÍTULO 5

IMPLEMENTACION DEL SOFTWARE DE VISIÓN ARTIFICIAL

En este capítulo se describirá la lógica de la aplicación desarrollada como parte del sistema de visión desarrollado en este Proyecto. Antes de describir en detalle la aplicación, se realizará un breve resumen de las herramientas empleadas para su implementación. También se describirá el espacio de trabajo sobre el que se ha desarrollado la aplicación, es decir, los distintos archivos y directorios que comprende, desde los códigos fuente hasta los archivos de entrada y salida.

La descripción del funcionamiento de la aplicación se va a dividir en varios apartados según los diferentes aspectos que la componen, captura y preprocesado de imágenes, comunicación con el sistema de control, calibración, y obtención de resultados. Dentro de cada uno de los aspectos se realizará un análisis del problema en donde se justificará la solución adoptada, e incluso a veces se enumerarán posibles alternativas interesantes. También se describirán brevemente los componentes implicados (clases, controles ActiveX,...), lo que permitirá posteriormente explicar los algoritmos implementados.

5.1. Herramientas empleadas para el desarrollo del software de visión

A la hora de desarrollar cualquier tipo de aplicación y en particular en el caso de las aplicaciones de visión, resulta crucial la elección del lenguaje de programación y del compilador adecuado. El compilador será el encargado de, a partir del código de alto nivel escrito en un determinado lenguaje de programación, generar el código máquina correspondiente que será ejecutado por el procesador. Así, la elección del compilador está condicionada por el lenguaje de alto nivel empleado. Para la elección del lenguaje de programación, se deben tener en cuenta muchos y muy distintos factores:

- Existencia previa de librerías ya implementadas en ese lenguaje que resuelvan parte de los problemas.
- Eficiencia exigida a la aplicación. Un lenguaje compilado es mucho más eficiente que uno interpretado.
- Portabilidad de la aplicación. Puede ser que sea necesario poder ejecutar la aplicación sobre distintos sistemas operativos instalados en máquinas con distintas arquitecturas sin tener que rescribir ni recompilar su código.
- Conveniencia del uso de un determinado paradigma de programación (estructurado u orientado a objetos).
- Etc.

Este Proyecto comprende varios problemas de elevada complejidad como son los concernientes a la visión artificial y la comunicación con el autómatas. Por tanto, para no hacer de este Proyecto una tarea increíblemente larga y difícil, se hace necesario el uso de librerías que resuelven gran parte de estos problemas. Gracias al uso de estas librerías en este Proyecto sólo habrá de centrarse en los aspectos novedosos que este implica. No obstante, existen librerías para solucionar estos problemas en varios lenguajes de programación, incluso distintas librerías con distintas características para el uso en un mismo lenguaje.

La eficiencia del lenguaje es un aspecto a considerar, debido a lo costosos computacionalmente que suelen ser los algoritmos de visión artificial. Aún así, esta no se trata de una aplicación que forme parte de un sistema al que se le exija una alta velocidad, hay que recordar que se va a usar para realizar estudios en el laboratorio y no para un proceso industrial.

La portabilidad de la aplicación implementada en este Proyecto no es un requisito esencial, ya que se construirá especialmente para la plataforma seleccionada, y no se espera que tenga que migrar a ninguna otra. Los lenguajes portables suelen ser interpretados y por lo tanto menos eficientes, sobre todo en lo que respecta a manejo de hardware.

Para la elección de la librería lo más importante a tener en cuenta es que contenga la mayor funcionalidad necesaria posible, y que sea compatible con el hardware y la plataforma utilizada. También va a resultar importante su funcionamiento, en sentido de eficiencia e integración con el compilador, de modo que esta facilite el acceso a sus componentes y su interfaz respecto al programador. Es también importante que la librería esté bien documentada, de manera que permita hacer un buen uso de ella y la rápida solución de problemas que surjan.

5.1.1. Lenguaje de Programación

Dadas las necesidades de tener un lenguaje que dispusiera de unas librerías de visión artificial y comunicación compatibles el hardware y la plataforma, que fuera eficiente, permitiera la gestión de los recursos, y tuviera la expresividad suficiente para implementar la aplicación en un tiempo razonable, se ha optado por emplear el lenguaje C++.

C++ se puede considerar una evolución de C. Lo que hace C++ es proporcionar soporte para la *Programación Orientada a Objetos* (POO) a C y mejorar algunas características. C++ es totalmente compatible con C, lo que quiere decir, que el código desarrollado en C++ puede aprovechar código existente en C. Además, intenta conservar las características que

proporcionaron el éxito a C: un lenguaje de alto nivel, pero que permite usar mecanismos de bajo nivel para aumentar la eficiencia del código.

El paradigma de POO ha surgido como la evolución de los enfoques de programación hacia una abstracción mayor, hasta la creación de tipos de datos que representan los elementos o entidades que intervienen en el problema, para así proporcionarle al programador herramientas para representar los elementos que existen en el *dominio del problema*. Entonces, en este enfoque todo es un objeto, es decir, cada componente conceptual del problema que se intenta resolver (cámara, imagen, servidor,...) es un objeto: el cual posee un estado, propiedades (en cada instante se encuentra en un posición definida), y comportamiento (reacciona ante estímulos de otros objetos).

Un programa según el enfoque de la POO es un conjunto de objetos que interaccionan entre sí (se envían mensajes) para conseguir un resultado. La ventaja de este enfoque es que permite reducir la complejidad de un problema ya que se van dividiendo y aislando los elementos que intervienen en él. Pero además, gracias al concepto de *herencia* se puede escribir código que puede ser utilizado de manera general por los objetos derivados, mientras que el *polimorfismo* asegura que se ejecutará el código adecuado mediante el mecanismo de *enlace dinámico*. La herencia junto con la *encapsulación* u ocultación de la información, permite que el código sea reutilizable y fácil de mantener y extender.

Las librerías utilizadas, ActiveMIL, pueden ser utilizadas desde C++ y Visual Basic. La elección de C++, pese a ser un lenguaje más complejo que Basic, se debió a la mayor eficiencia (en cuanto a tiempo de ejecución) de las aplicaciones escritas C++, dado su carácter de código compilado (Basic es un lenguaje interpretado). Además, C++ ofrece todas las herramientas de la POO tales como la herencia, encapsulación y polimorfismo, mientras que Visual Basic simplemente es *basado en objetos*, lo que le permite el uso de objetos pero no de conceptos tales como la herencia y polimorfismo.

5.1.2. Librerías de Visión Artificial

Como ya se ha explicado con anterioridad, el uso de librerías simplifica enormemente el coste y el esfuerzo de desarrollo de las aplicaciones de visión artificial. En este Proyecto se ha utilizado la librería ActiveMIL (ActiveX Matrox Imaging Library) en su versión 8.0, producto de la compañía Matrox® [Web 4].

ActiveMIL es un conjunto de controles ActiveX™ basado en la librería de procesamiento de imagen de Matrox®, MIL. ActiveMIL fue diseñada para poder ser usada desde herramientas RAD (Rapid Application Development), tales como Microsoft® Visual Basic .NET, ó Microsoft® Visual C++ .NET, que a la postre ha sido el compilador usado en este Proyecto para el desarrollo del software de visión artificial. Un control ActiveX (y por tanto ActiveMIL) no es más que un objeto [Web 10] según describe la el paradigma de la POO, poseyendo un estado, propiedades y un comportamiento (métodos).

La utilización de ActiveMIL, no solo se debe a que permiten el manejo del hardware de adquisición de imágenes (del mismo fabricante que las librerías), además de la captura de imágenes desde él, sino también a que esta basa en las MIL. El grupo de investigación DSIE, en cuyo seno se ha desarrollado este Proyecto, ha utilizado las MIL en el desarrollo de otros

sistemas visión con excelentes resultados, luego se disponía de un buen referente en cuanto a su funcionamiento.

ActiveMIL es independiente del hardware, o sea que no sólo permite el funcionamiento sobre una tarjeta de Matrox®, también puede funcionar sobre cualquier controlador gráfico soportado por Windows. ActiveMIL, en su versión completa (la utilizada en este Proyecto) permite grabar imágenes, mostrarlas y escribir sobre ellas, además un gran conjunto de operaciones de procesado y análisis de imágenes, todo ello agrupado sobre un juego de controles ActiveX. Las operaciones de análisis incluyen: medidas, análisis de *blobs*, calibración, reconocimiento de caracteres (OCR), búsqueda de patrones, búsqueda de contornos, y operaciones de lectura/escritura de códigos 2D. Las operaciones de procesado de imágenes incluyen operaciones punto a punto, estadísticas, filtrado espacial, morfológicas, transformada rápida de Fourier (FFT), y operaciones geométricas [8]. En general, ActiveMIL puede:

- Grabar imágenes en escala de grises de hasta 16-bit, o imágenes en color.
- Procesar imágenes formadas por enteros de 1, 8, 16, y 32-bit, ó coma flotante.
- Procesar imágenes en color dependiendo de la operación. Cada banda de una imagen en color es procesada individualmente, una después de la otra. Las operaciones estadísticas y de análisis no soportan imágenes en color.
- Mostrar imágenes en escala de grises de 1, 8, o 16-bit, ó imágenes en color.

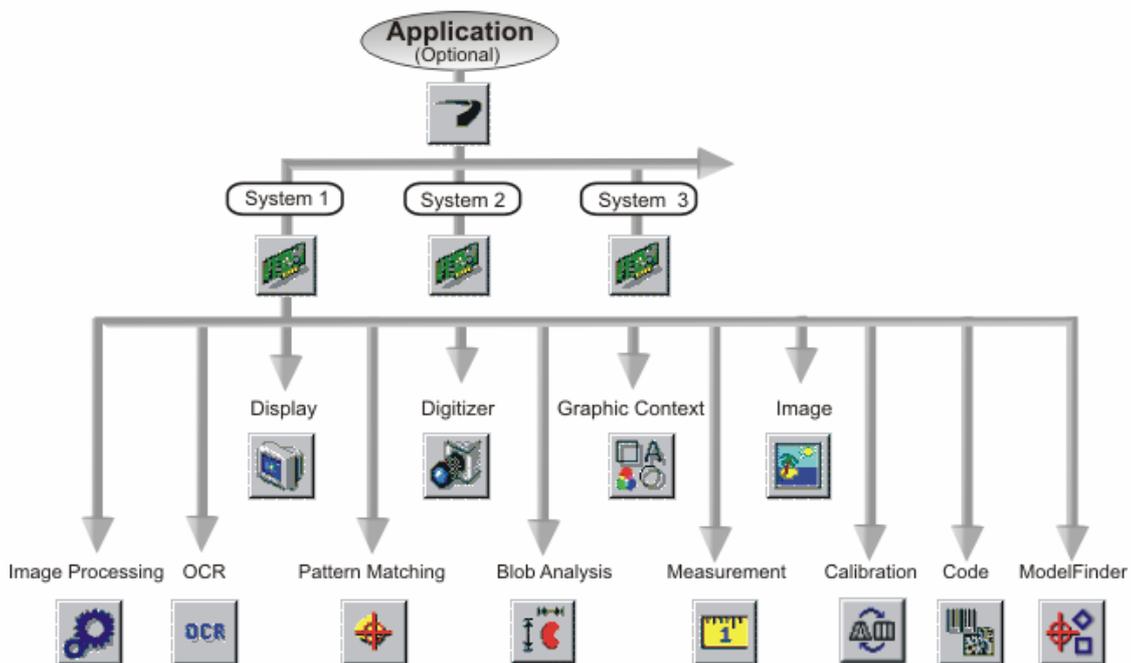


Figura 5. 1. Controles de ActiveMIL y ejemplo de su posible organización en una aplicación.

El uso de ActiveMIL en lugar de MIL, se debe a que ActiveMIL ha sido diseñada para utilizarse en herramientas de desarrollo de interfaces gráficas de usuario, lo que ha facilitado el desarrollo de aplicaciones de visión artificial que poseen interfaces amigables para el usuario (*user friendly*). Esta era una característica que se deseaba conferir a la aplicación que controlase AUTOMAN-TUNEL. En contraposición, las MIL ofrecen un control a un nivel un poco más bajo de las

operaciones y del hardware, no obstante, con ActiveMIL se puede desarrollar una aplicación de visión artificial con la eficiencia necesaria para las necesidades de este Proyecto.

El mayor inconveniente de ActiveMIL, radica en la necesidad de licencia para el desarrollo y ejecución de aplicaciones. Esto eleva el coste del sistema, aunque se ha podido permitir ya que en el DSIE se disponen de varias de estas licencias. Por supuesto, también son cerradas y no se puede acceder a su código, lo que a veces impide ajustar exactamente cada algoritmo a las necesidades de ese momento.

5.1.3. El Compilador

La transformación del código escrito en un lenguaje de alto nivel a código máquina corre a cargo de un compilador. La elección del compilador depende del lenguaje de alto nivel utilizado y de la plataforma sobre la que se desee ejecutar el código empleado.

La implementación de la aplicación relacionada con la parte de visión artificial en este Proyecto se ha realizado en C++, y la plataforma donde se ejecutará tendrá instalado el sistema operativo Windows XP. Existen multitud de compiladores para C++ que pueden ser utilizados sobre Windows XP. Sin embargo, dado que ActiveMIL ha sido desarrollada para ser utilizada sobre Visual .NET y dispone de plantillas, documentación y un conjunto de ejemplos, que facilitan su uso sobre esta herramienta, ésta ha sido la utilizada.

Visual C++ .NET, la herramienta para C++ que provee Visual Studio .NET, más que un compilador es un entorno de desarrollo de aplicaciones C/C++. Viene acompañado de una extensa librería de ayuda al programador (MSDN), que incluye multitud de ejemplos y referencia a las funciones de las librerías estándar. Visual C++ .NET permite el desarrollo de ejecutables, de librerías que cumplan el estándar DLL (Dinamic-Link Library), librerías estáticas, *Makefiles*, etc. La aplicación desarrollada en este proyecto se hará en forma de ejecutable.

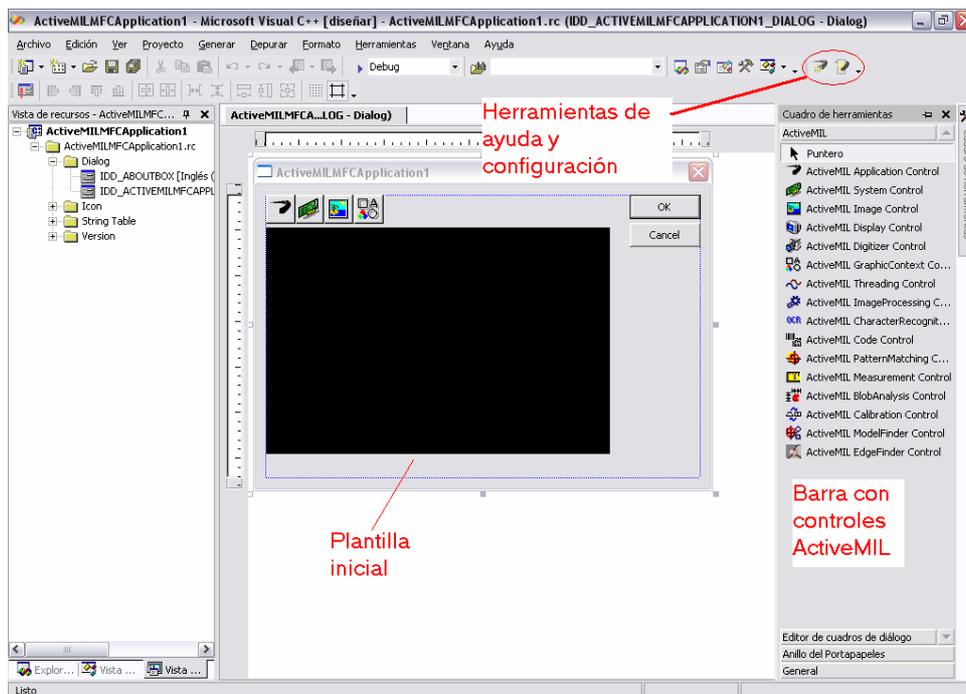


Figura 5. 2. Visual C++ .NET con los complementos de ActiveMIL.

ActiveMIL dispone de un complemento o *Add-In* para Visual C++ .NET, de manera que al ser instalado desde Visual C++ .NET se pueden realizar aplicaciones que utilizan ActiveMIL a partir de plantillas, además de poderse acceder desde el compilador a un barra en donde aparecen todos los controles de ActiveMIL, a la ayuda y documentación que provee ActiveMIL, y a la herramienta de configuración de ActiveMIL (ver figura 5.2).

En Visual C++ .NET se trabaja con soluciones (*solutions*), las cuales pueden ser vistas como espacios de trabajo. Una solución es un contenedor para proyectos (*projects*) y sus respectivas configuraciones. Un proyecto se define como una configuración y un grupo de archivos, escritos en lenguajes de programación, que generarán tras su compilación un archivo binario. Una solución puede contener múltiples proyectos, entre los que pueden existir (o no) interdependencias, aunque en este Proyecto la aplicación de visión artificial esta contenida en un solo proyecto.

5.2. El Espacio de Trabajo

Dado que para el desarrollo de la aplicación de visión artificial se ha utilizado Visual C++ .NET, ésta se ha construido en forma de un proyecto que contiene una serie de archivos fuente, con sus dependencias, y una configuración que permite la correcta generación del ejecutable. En este Proyecto se utiliza código de desarrollo propio (escrito en C++ e IDL), de ActiveMIL, MFC (*Microsoft Foundation Classes*) y librerías estándar.

El código encargado de lanzar la aplicación está contenido en una clase llamada *ScanTunnelApp*. Hay que resaltar que en C++ una clase necesita de dos archivos, uno con cabecera *.h que especifica la definición de la clase, y otros *.cpp en donde se implementa su funcionalidad. Esta clase contiene un método que es el encargado de crear un *thread* para la ejecución del interfaz gráfico, a través del cual se va a poder acceder a toda la funcionalidad de la aplicación, además de crear e inicializar a la clase que contendrá el interfaz gráfico *ScanTunnelDlg*.

El cuerpo principal de la aplicación está contenido en la clase *ScanTunnelDlg*, la cual define un interfaz gráfico compuesto por un diálogo. El diálogo contiene botones a través de los cuales se puede acceder a toda la funcionalidad de la aplicación, además sobre él se muestran las imágenes resultado. Esta clase va a incluir otras clases, estructuras de datos y definiciones, tales como la clase *ServerOPC*, diseñada en este Proyecto para comunicarse con un servidor OPC, además de clases, funciones y definiciones de ActiveMIL, MFC, y librerías estándar.

La compilación de *ServerOPC* necesita de una definición previa del contrato (qué y cómo se va a transmitir) entre esta clase y un servidor OPC. Este proceso se realiza a través de un archivo IDL en donde se definen los interfaces y tipos de datos que *ServerOPC* va a poder utilizar. A partir del archivo IDL (*OPC.idl*) el compilador genera una serie de archivos en C (*.c y *.h) que se pueden usar como *stub* en las llamadas a procedimiento remoto con el servidor OPC de Siemens® que se ha utilizado.

Los componentes de MFC, además de una serie de definiciones sobre la plataforma, son incluidos a través de un archivo de cabecera llamado "*stdafx.h*". El archivo de cabecera "*resource.h*" contiene la definición de los identificadores de los controles ActiveX incluidos en el interfaz gráfico, además de la información obtenida del editor del gráfico de Visual C++ que

permite definir las características del interfaz gráfico. En la implementación de ScanTunnelDlg se establece un vínculo con el control gracias a su identificación, gracias al cual se pueden sincronizar durante la ejecución las propiedades de los controles con las variables de la clase ScanTunnelDlg.

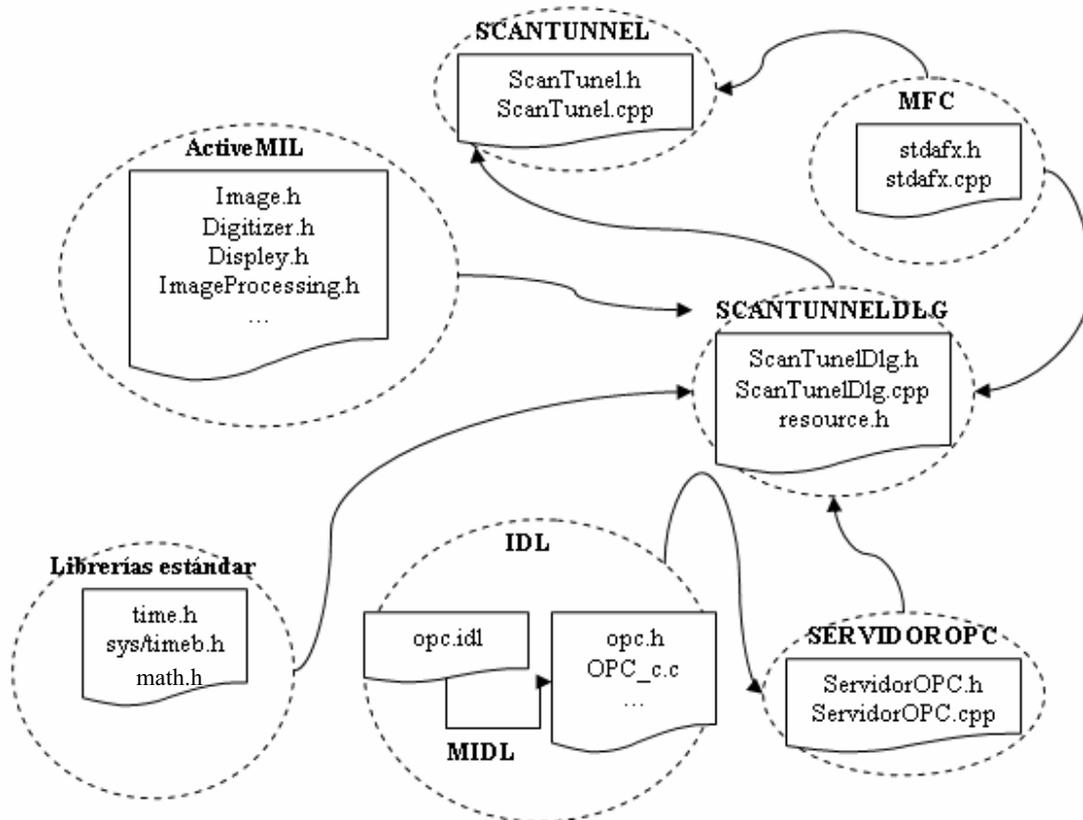


Figura 5. 3. Esquema las relaciones básicas entre el código implementado para visión artificial.

5.3. Captura y Preprocesado de imágenes

En este apartado se describe la parte del software de visión artificial desarrollado encargada de la adquisición y preprocesado de imágenes. Esta funcionalidad se corresponde con las etapas 1 y 2 del modelo de sistema de visión por computador explicado en el capítulo 2. La construcción de esta funcionalidad se ha servido de varios de los controles que ofrece ActiveMIL.

5.3.1. Análisis del Problema

La primera etapa del problema consiste en la adquisición de imágenes desde la tarjeta de adquisición. Esta parte es sencilla una vez fijado el modelo hardware explicado en el apartado 4.3, ya que ActiveMIL ofrece controles ActiveX que permiten controlar todas las tarjetas de adquisición de Matrox®, y las cámaras FireWire de Basler®.

El preprocesado de imágenes sin embargo, ofrece una problemática mayor. Como se expone en el apartado 4.2.1, debido a como trabaja el sistema de iluminación, la cámara a de trabajar en régimen de máxima ganancia, lo que va a condicionar la inclusión de mucho ruido en las

imágenes. Este ruido hace inviable el tomar imágenes con exposiciones prolongadas que permitan hacer un correcto promediado del flujo de líquido que atraviesa el plano iluminado, ya que el ruido enmascararía la luz reflejada por las gotas de líquido.

Para resolver el problema del ruido se ha optado por hacer muchas capturas de imágenes, tantas como se necesite para hacer un correcto promediado del flujo, con tiempos de exposición de la cámara muy cortos. En exposiciones cortas el ruido que aparece en las imágenes es mucho menor, con lo que es más fácil de aislar la información interesante. El problema es que la información proveniente de las gotas de líquido también va a ser mucho menor, con lo que con una imagen no se va a obtener información suficiente sobre el flujo de líquido. Por esto, es necesario integrar todas estas imágenes, una vez se les ha extraído el ruido, para así poder obtener un promedio suficientemente representativo sobre el flujo de líquido que atraviesa el plano de iluminación.

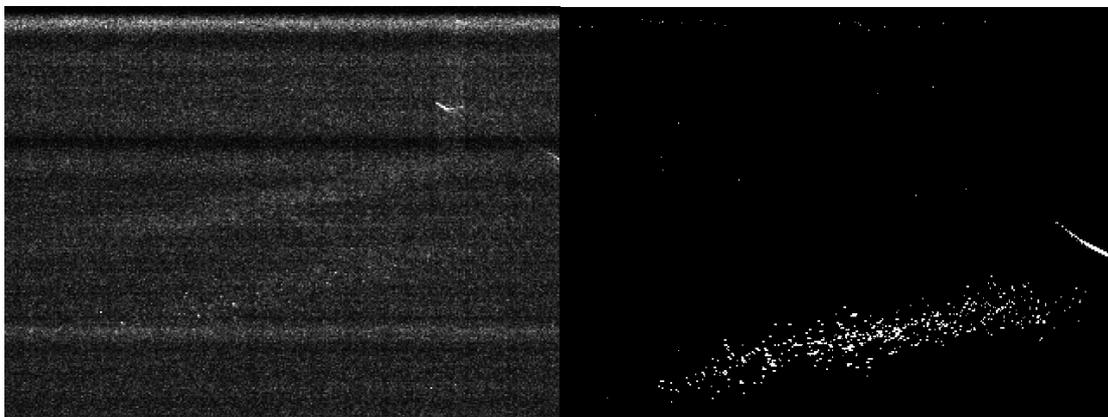


Figura 5. 4. Imagen de un corte de la nube de líquido con una captura de exposición larga (izquierda). Imagen resultado del proceso de integración de capturas cortas a las que previamente se ha eliminado el ruido (derecha), obsérvese se ha aislado la información del flujo de líquido.

5.3.2. Controles ActiveMIL utilizados

En la implementación de la funcionalidad de adquisición y preprocesado de imágenes, el software de este Proyecto se ha servido de varios de los controles ActiveX que ActiveMIL provee.

Para comenzar a capturar lo primero que se necesita importar es un control que represente al hardware de adquisición, en este caso la tarjeta Meteor-II/1394 instalada en el PC. Aunque también se puede utilizar para acceder a un controlador gráfico, o la CPU local. Es necesaria la inclusión de este control para poder añadir el resto de controles ActiveMIL que se han utilizado.

La grabación de imágenes necesita también de un control que represente a la cámara (*Digitizer*), y en el que se identifique la dirección de las imágenes generadas por la cámara en la tarjeta de adquisición. A través de este control se pueden definir los parámetros de grabación de la cámara de manera sencilla, sobre todo en el caso particular de la cámara utilizada en este Proyecto gracias al interfaz IEEE 1394 que utiliza. Mediante un método que provee el controlador se pueden obtener las imágenes aisladas (*frames*), según el formato que se haya especificado y según los parámetros fijados para la cámara. Para este caso se grabarán imágenes en color (con 3 bandas), de tamaño 640x480 píxeles, utilizando la mayor ganancia posible de que dispone el sensor de la cámara y con un periodo de exposición de 33.3ms.

El resultado de la grabación se almacena en un control específico para contener imágenes (*Image*). Este control va a contener imágenes permitiendo realizar operaciones sobre ellas, incluso implementa funcionalidad para poder cargar y grabar imágenes en (o desde) archivos en varios formatos estándares como BMP, TIFF o JPEG, ó en el formato propio de Matrox®, MIM. Se han necesitado tres de estos controles, uno en el que se aloja la imagen capturada desde el hardware de adquisición y que posee el formato que específico en el control *Digitizer*. Esta imagen posee las tres bandas RGB, pero a la aplicación sólo le interesa la banda R (ver apartado 4.2.1) que será la que contendrá la información de la banda roja en donde emite la fuente de iluminación utilizada, ésta se obtiene mediante otro control *Image* derivado (*Child Image*) del anterior, pero que sólo direccionará la banda roja. Por lo que a final se obtiene una imagen en escala de grises de 8-bit.

Para la eliminación del ruido en las imágenes capturadas se ha aprovechado la funcionalidad que ofrece el control *ImageProcessing*. Este control contiene métodos que permiten realizar operaciones punto a punto, estadísticas, de filtrado espacial y morfológicas sobre imágenes. Otro control *ImageProcessing* ha sido incluido para realizar la operación de integración de imágenes. En principio solo habría sido necesario un control *ImageProcessing* para realizar la eliminación de ruido y suma de imágenes, al que durante la ejecución se le fueran cambiando las propiedades de imágenes de partida y resultado. No obstante, en aras de un menor tiempo de ejecución se ha decidido emplear dos controles con propiedades fijas, en lugar de ocupar tiempo de ejecución en operaciones no directamente implicadas en el procesado de imágenes. En el proceso de adquisición y preprocesado de imágenes, puesto que junto con el de control del sistema de escaneado, limita el tiempo de ejecución del sistema, se le va a dar preferencia a la eficiencia en tiempo en detrimento del ahorro de memoria.



Figura 5. 5. Controles ActiveMIL empleados en la captura y preprocesado de imágenes, de arriba a abajo y de izquierda a derecha *System*, *Digitizer*, *Image* e *ImageProcessing*.

5.3.3. Proceso de Adquisición y Preprocesado de Imágenes

Para cada corte del sistema de iluminación, la aplicación va a extraer una imagen en la que se promedia el flujo de gotas de líquido que pasan por el plano que define la fuente de iluminación. El proceso de extracción de esta imagen se basa en el siguiente algoritmo:

1. Repetir 100 veces
 - 1.1. Grabación de un frame (imagen color)
 - 1.2. Extracción de la banda R (imagen escala de grises)
 - 1.3. Eliminar ruido (umbralización)
 - 1.4. Sumar imagen escala de grises con la imagen almacén
2. Retornar resultado (imagen almacén)

Figura 5. 6. Algoritmo de adquisición y preprocesado de imágenes.

La luz reflejada por las gotas de líquido (información relevante) genera en la imagen puntos de unos pocos píxeles de alta iluminación, y en cambio el ruido se muestra como puntos de uno o

unos pocos píxeles poco luminosos. Luego una forma sencilla de eliminar el ruido en cada captura consiste en el umbralizado de la imagen, de manera que la información relevante se tome la máxima intensidad y el ruido que para casi todos los casos no superará el umbral se borra. Existen otros métodos para la eliminación de ruido, como son el suavizado mediante filtros, ó mediante el empleo de operaciones morfológicas. Sin embargo, dada la naturaleza de la información y del ruido la umbralización presentaba mejores prestaciones, ya que por ejemplo mediante operaciones morfológicas se eliminaba también los puntos de información relevante puesto que estos poseían un tamaño similar al del ruido.

El control ImageProcessing puede establecer automáticamente el nivel de umbralización a partir del histograma de la imagen. Sin embargo, el algoritmo que implementa solo trabaja correctamente si el histograma de la imagen presenta dos máximos bien diferenciados, lo cual no ocurre en este caso (ver imagen 5.7). Por esto, se ha utilizado un umbral fijo (que el usuario puede modificar a través del interfaz gráfico), lo cual es robusto debido a que el aislamiento lumínico del cajón del túnel mantiene estables las condiciones de iluminación. La experiencia acumulada a lo largo de todas las pruebas hechas durante el desarrollo del Proyecto nos indican que el umbral más adecuado, es el que tiene valor 12. Además, el no utilizar el algoritmo de búsqueda de umbral mejora el tiempo de ejecución del proceso.

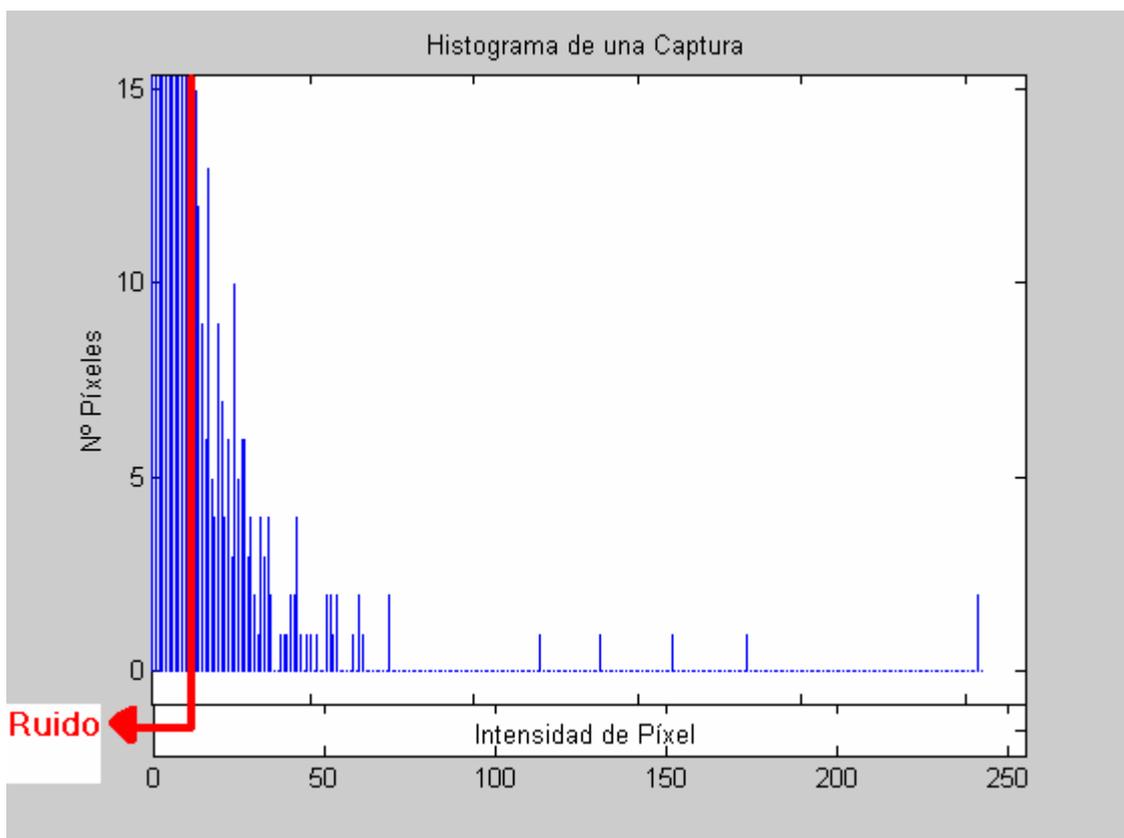


Figura 5. 7. Histograma de una captura de 33.3ms.

Según el algoritmo anterior en el control *Image* que se utilice como almacén se acumula el resultado de la suma aritmética píxel a píxel de todas las capturas realizadas una vez se les ha extraído el ruido. Así el tiempo real de promediado de flujo de gotas de líquido, se corresponde con la suma del tiempo de exposición empleado en las cientos de capturas realizadas, según la

cantidad de capturas que se defina el tiempo de promediado puede alcanzar el valor de segundos.



Figura 5. 8. Imagen resultado de una sola captura (3 puntos luminosos), imagen resultado de la integración de 100 capturas (0.33seg).

5.4. Comunicación con el Autómata

En el apartado anterior se ha visto el proceso de adquisición y preprocesado de imágenes, de manera que para un corte en particular se obtiene la distribución de gotas de líquido que ha atravesado el corte. Pero para obtener este resultado para todos los cortes que se desea, se hace imprescindible establecer una comunicación con el sistema de escaneado que es el encargado de controlar la posición de la fuente óptica.

El sistema de visión artificial debe enviarle información al autómata, que es quien controla el sistema de escaneado, para indicarle cuando se arranca el proceso de escaneado, y cuando ha terminado de capturar imágenes de un corte y debe de pasar a otro. Además, debe vigilar el estado del autómata para saber cuando la fuente de iluminación se ha estabilizado sobre un corte para poder comenzar a capturar imágenes, ó saber que se ha terminado el proceso de escaneado.

5.4.1. Análisis del Problema

Se desea establecer una comunicación con un dispositivo remoto (autómata), conectado al PC sobre el que se está ejecutando la aplicación de visión artificial, a través de un enlace PROFIBUS-DP. El PC dispone de una tarjeta Siemens® CP 5611, insertada sobre PCI, con el interfaz PROFIBUS-DP, luego el problema se resume en establecer comunicación con esta tarjeta.

Para resolver la comunicación con la tarjeta (o módulo PROFIBUS) se han planteado dos opciones distintas. La primera de ellas consistiría en construir un *driver* específico para la aplicación de visión artificial, que comunicase con el módulo vía PCI. La segunda opción, y por la que finalmente se ha optado, consiste en comunicarse con el módulo a través de un intermediario en forma de servidor OPC. Las características y ventajas en general del uso del servidor OPC ya se describen en el apartado 4.5.3.

La opción del servidor OPC, en principio, podría parecer menos eficiente para este Proyecto que el desarrollo de un driver específico, debido a que los datos han de pasar a través de un intermediario. No obstante, el modelo OPC ofrece a este sistema una serie de ventajas:

- Comunicación eficiente: el acceso al módulo hardware lo realiza el servidor OPC que ha sido implementado por el mismo fabricante del módulo, lo que garantiza un acceso eficiente y robusto.
- Uso de estándares: el interfaz que debe de implementar y utilizar la aplicación de visión artificial, OPC, es estándar, y está bien documentado y extendido.
- La comunicación es independiente del hardware: se puede cambiar el módulo de comunicación ó el autómeta si tener que modificar el código de la aplicación de visión.
- Control de alarmas: OPC provee mecanismos para que el cliente (aplicación de visión) sea notificado de la ocurrencia de acontecimientos y de condiciones de alarmas especificadas.
- Acceso a datos históricos: OPC permite la lectura, procesamiento y corrección de datos históricos con un eficiente motor de acceso.

Una de las mayores ventajas que ofrece OPC no afecta mucho a este Proyecto, pero si ha sido tomada muy en cuenta, se trata de la posibilidad de que otras aplicaciones puedan tener acceso concurrentemente al módulo de comunicación. Dos paquetes, generalmente, no pueden acceder simultáneamente al mismo dispositivo, puesto que cada uno contiene drivers independientes. El uso de OPC va a tener una trascendencia crucial cuando se vaya a construir el prototipo AUTOMAN, ya que va a permitir que el sistema AUTOMAN-TUNEL y el prototipo AUTOMAN puedan compartir PC, bus y autómeta, funcionando concurrentemente. Es más el software implementado para realizar la comunicación desde una aplicación C++ a través de OPC (clase *ServerOPC*), puede ser reutilizada por la aplicación que vaya a controlar al prototipo AUTOMAN desde el PC.

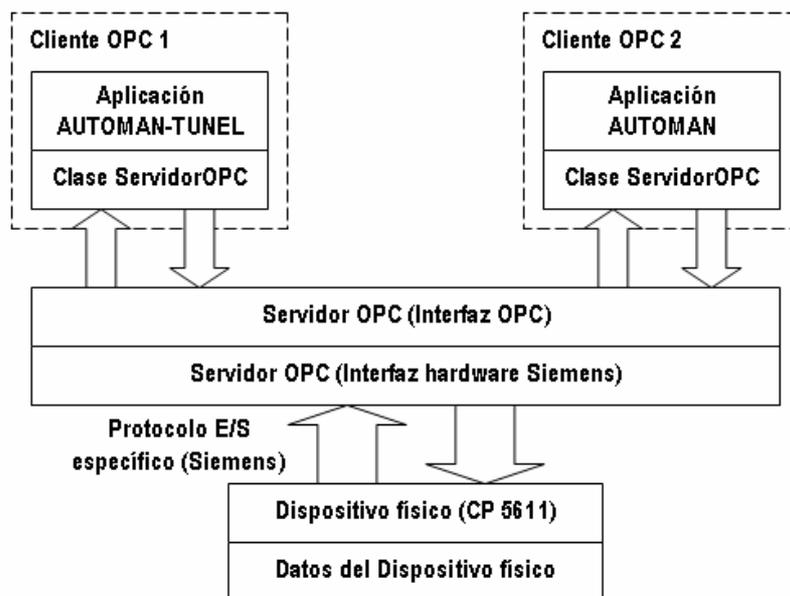


Figura 5. 9. El uso OPC posibilitará que la aplicación de visión de AUTOMAN-TUNEL y la futura aplicación de control del prototipo AUTOMAN, puedan funcionar concurrentemente usando el mismo hardware de comunicación.

5.4.2. La Clase ServerOPC

La comunicación entre la aplicación de visión, que actúa como cliente, con el servidor OPC se ha solucionado mediante la implementación de la clase ServerOPC, escrita en C++ al igual que la aplicación de visión. Así, la aplicación de visión instanciará un objeto de esta clase, a través del cual podrá comunicarse con el servidor OPC, lo que implica hacerlo también con el autómeta vía módulo PROFIBUS, gracias a la funcionalidad que ofrece esta clase.

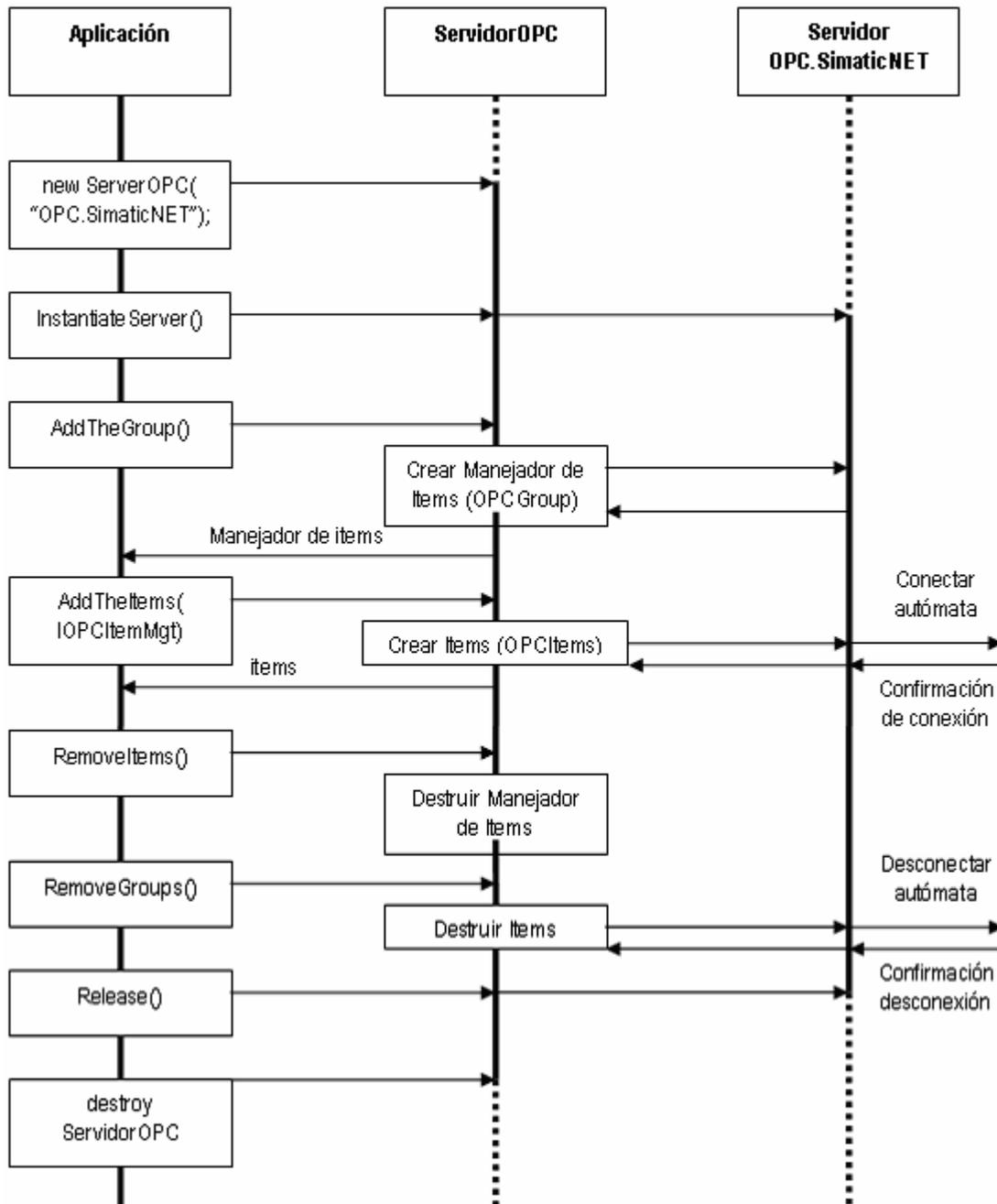


Figura 5. 10. Creación, inicialización y destrucción de los objetos de la clase ServerOPC.

El interfaz que esta clase ofrece al usuario (aplicación) es independiente de éste, luego puede ser utilizado por cualquier aplicación escrita en C++. Además, el interfaz con el que se comunica

con el servidor OPC también es estándar, lo que significa que funcionará sobre cualquier servidor OPC, en consecuencia con cualquier hardware, de cualquier fabricante, y no está restringida al uso junto al hardware/software de Siemens® utilizado en este Proyecto. Esto ofrece la posibilidad de poder utilizar esta clase para resolver problemas de comunicación con dispositivos de automatización en otros proyectos.

Lo que la clase ServerOPC hace es encapsular los tres objetos que intervienen en OPC, *OPCServer*, *OPCGroup* y *OPCItem*. *OPCGroup* representa a un proceso individual de variables, *OPC items*, *OPCItem* representa a uno de los items de ese grupo. En este proyecto los items se corresponderán con direcciones de la perifería distribuida PROFIBUS-DP, que el PC y el autómatas conectados por el bus comparten.

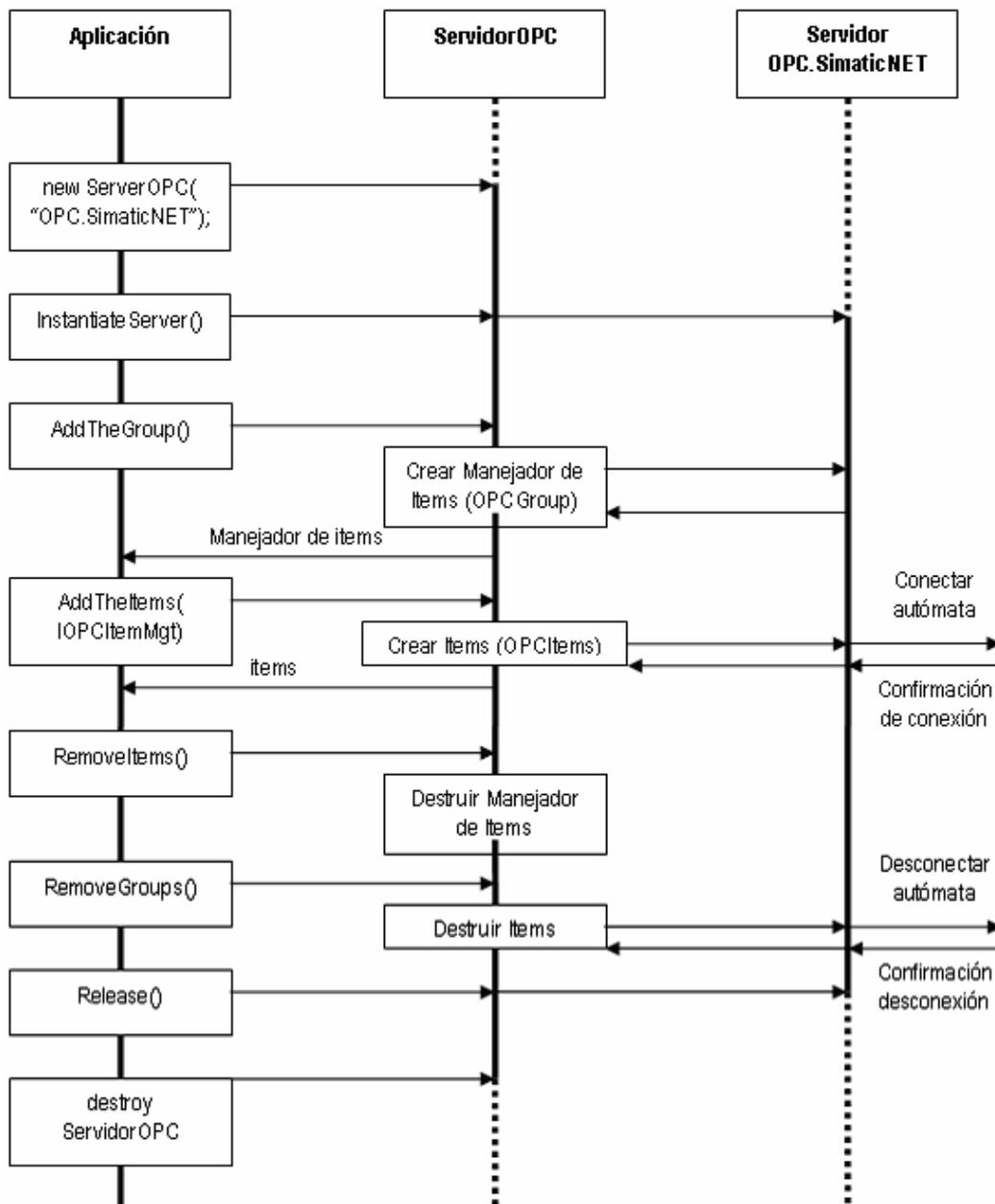


Figura 5. 11. Creación, inicialización y destrucción de los objetos de la clase ServerOPC.

Gracias a la clase ServerOPC el usuario tan solo tiene que pasar el nombre del servidor OPC al que se quiere conectar (*OPC.SimaticNET*), a través de llamada a sus métodos crea los objetos OPC necesarios para la comunicación, y devuelve los interfaces mediante los cuales se puede acceder a los ítems y a los valores que éstos contienen.

Mediante ServerOPC se pueden leer y escribir datos en los ítems mediante una sola llamada a un método, lo que significa hacerlo en la periferia distribuida del bus PROFIBUS a la que también puede acceder el autómatas. La lectura y escritura sobre los ítems se realiza de forma síncrona, es decir, el método de escritura/lectura de ServiorOPC no devuelve el control a la aplicación hasta que haya recibido una respuesta del módulo hardware a través del servidor OPC. A los métodos de lectura/escritura se le han de pasar los interfaces de los objetos OPCGroup y OPCItem, para saber sobre que ítem actuar, esto se ha hecho así para que se puedan leer ó escribir varios ítems a la vez.

5.4.3. El Proceso de Escaneado

En el apartado 5.3 se ha visto como la aplicación de visión genera una imagen sin apenas ruido del flujo de gotas de líquido que atraviesan el plano de iluminación. Puesto que se quiere obtener imágenes sobre toda la nube, la aplicación de visión a de indicarle al autómatas cuando ha terminado de generar una imagen de un corte y debe cambiar la altura de la fuente de iluminación. Además, debe sondear al autómatas para saber cuando se ha estabilizado y no comenzar a tomar imágenes con la fuente de iluminación en movimiento. La comunicación con el autómatas la realiza a través de un objeto de la clase ServerOPC vista en el apartado 5.4.3. Entonces para la correcta consecución del proceso de escaneado, entre el autómatas y el PC se establece un protocolo que se resume en el esquema de la figura 5.12.

Las señales de comunicación entre la aplicación de visión y la de control que corre sobre el autómatas son números enteros, los cuales codifican los distintos estados de la máquina de estados finitos que modela la aplicación de control (ver apartado 6.2.3). Así, las señales que van desde la aplicación de control a la de visión le indican a esta última en que estado se encuentra la aplicación de control, y las señales en sentido contrario expresan el estado sobre el que la aplicación de visión desea que se sitúe la de control.

Debido a que puede ser que la aplicación de control no alcance el estado solicitado por la aplicación de visión en un momento dado, la aplicación de visión vigilará el estado de la de control esperando que alcance el estado solicitado solo durante unos segundos, lo que en principio debería ser tiempo suficiente. De esta forma, se evita el bloqueo de la aplicación de visión cuando exista un fallo en sistema de escaneado. Los fallos en el sistema de escaneado podrían deberse a errores en la conexión, errores al leer los sensores, ó que el autómatas estuviera desconectado debido a un error en la puesta en marcha por parte del usuario.

En este Proyecto el proceso de escaneado consta de 4 paradas, ya que se ha valorado que son suficientes 4 cortes para ofrecer con bastante precisión información sobre la forma de la nube de líquido, además así el proceso de escaneado no consume un tiempo excesivo, unos 20 segundos (depende del tiempo de promediado que se desee). Si se quisiera aumentar el número de paradas habría que modificar ligeramente la aplicación de visión (PC) y la de control (autómatas). Aumentar el número de paradas supone aumentar el tiempo que dura el proceso de escaneado.

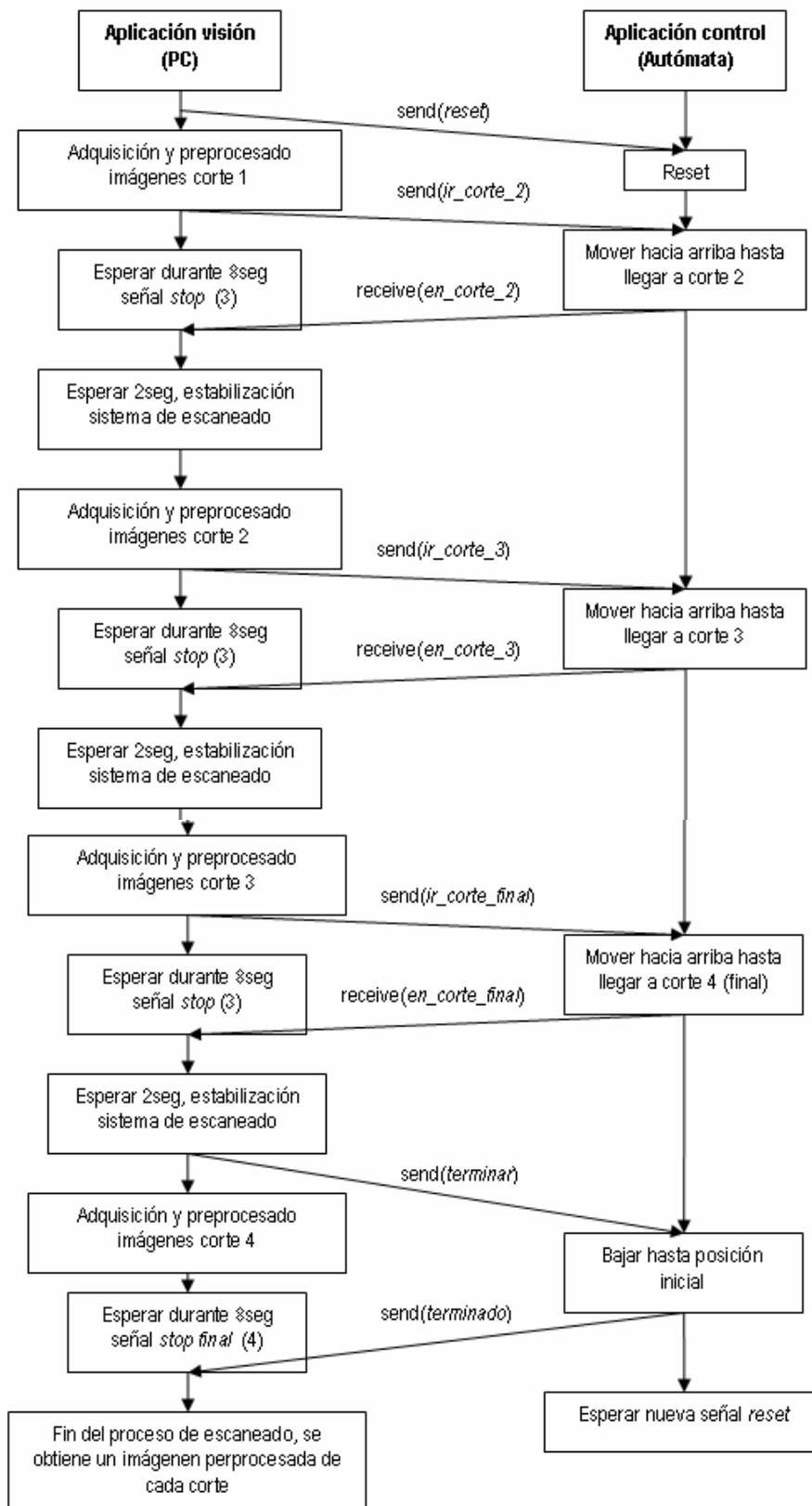


Figura 5. 12. Diagrama que muestra la comunicación entre aplicaciones durante un proceso normal de escaneado.

5.5. Calibración

Del proceso de escaneado se obtiene una imagen por cada corte con información sobre el flujo de gotas de agua de la nube de líquido. Pero es necesario tener calibrada la cámara para poder obtener resultados en unidades métricas y referidas al sistema de coordenadas de los objetos.

5.5.1. Análisis del Problema

Las imágenes se toman desde una cámara colocada en perspectiva con respecto al plano de iluminación de cada corte, es decir, el plano principal de la cámara y el de iluminación no son paralelos, luego la proyección un mismo objeto (conjunto de gotas) sobre la imagen varía en función de su posición en el plano de iluminación. Es por tanto necesario realizar una transformación de las imágenes, de manera que se corrija la deformación de las imágenes debido a la colocación en perspectiva de la cámara.



Figura 5. 13. Imagen de una rejilla rectangular de puntos tomada con la cámara en perspectiva, debido a esta colocación el rectángulo está deformado. En la derecha se muestra la imagen resultado de la transformación que corrige la perspectiva, mostrando la rectangularidad de la rejilla.

La transformación en perspectiva se reduce a encontrar las matrices de rotación y translación entre el espacio de los objetos (*real-world coordinates*) y el de la cámara, es decir, los parámetros extrínsecos a la cámara, matrices de rotación R , y de traslación T . Lo que permite la transformación de coordenadas entre espacios. Mediante una rotación se alinean los ejes de los espacios, y mediante una translación se igualan los orígenes entre los espacios:

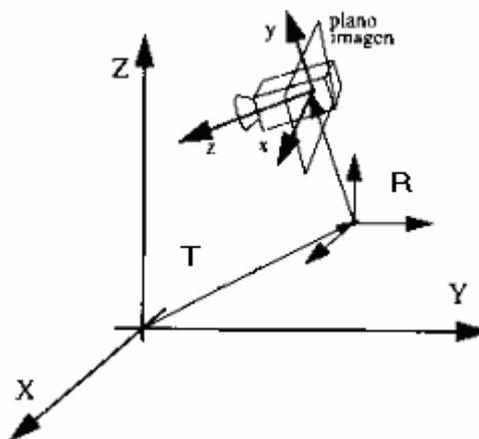


Figura 5. 14. Relación entre los sistemas de coordenadas del espacio de los objetos y de las imágenes.

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (5.1) \quad \mathbf{T} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (5.2)$$

También es necesario poder convertir los posibles resultados que se obtengan en sus unidades de medida propias. Es decir, los algoritmos de extracción de características de los objetos de las imágenes devuelven como resultado posiciones de píxeles o cantidades de píxeles, sin embargo en el espacio de los objetos (en donde se sitúa la nube de gotas) las cosas se miden en centímetros, metros, pulgadas, etc. Luego, también es necesario conocer la correspondencia entre un píxel en la imagen y el espacio real que este representa en plano de los objetos medido en la unidad métrica que se vaya a utilizar, en el caso de este Proyecto centímetros.

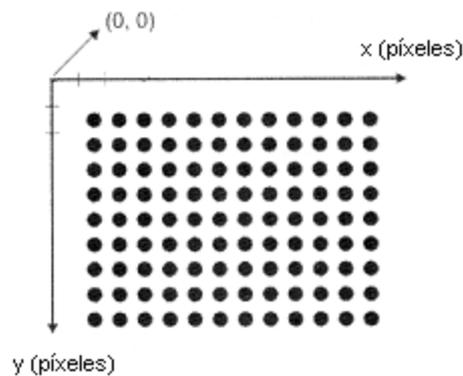


Figura 5. 15. Sistema de coordenadas para las imágenes utilizado por ActiveMIL.

Para realizar este segundo paso es necesario conocer los parámetros intrínsecos de la cámara, esto es, (u_0, v_0) que son las coordenadas de los puntos principales, α y β son el factor de escala para cada eje, y γ representa el ángulo entre los ejes de la imagen [5] (ver imagen 5.16). Estos valores se recogen en la matriz de parámetros intrínsecos de la cámara:

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.3) \quad \gamma = \alpha \cot \theta \quad (5.4)$$

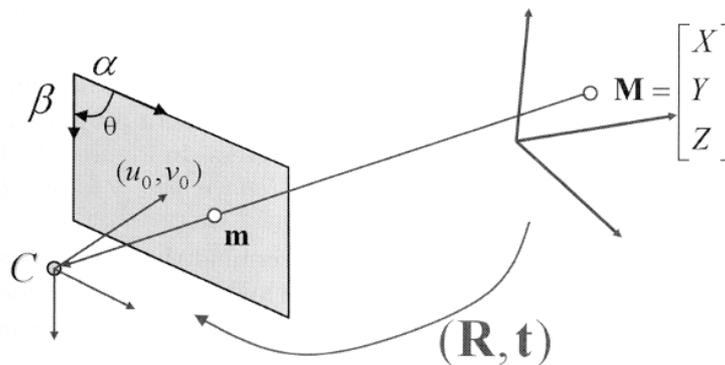


Figura 5. 16. Parámetros intrínsecos y extrínsecos de la cámara.

Una vez conocidos los parámetros intrínsecos y extrínsecos (P), ya se puede hacer una proyección de los puntos de la imagen (m) sobre el espacio objeto (M) (ecuación 5.5), mediante una expresión parecida también sería posible realizar el paso contrario. Luego, ya es posible realizar medidas en unidades métricas sobre el espacio objeto.

$$m = A[R \quad T]M = PM \quad (5.5)$$

Los parámetros para la definición se podrían refinar si se tuviera en cuenta también la distorsión que introduce la óptica de la cámara. Esta distorsión se suele modelar mediante tres tipos; distorsión radial, por descentrado y de prisma delgado. En este Proyecto estos tipos de distorsión no se va a considerar debido a que la óptica de la cámara posee una buena calidad proporcionando imágenes con poca distorsión, además la relación entre apertura y distancia focal es suficientemente pequeña para ayudar a garantizar imágenes con baja distorsión radial. Si la cámara hubiera tenido una apertura mayor ó una menor distancia focal quizá si hubiera sido necesario tener en cuenta la distorsión radial. Luego la calibración ha influido también en la elección de los parámetros de la óptica de la cámara, estableciendo un máximo para la apertura y un mínimo para la distancia focal.

Mediante este procedimiento se realiza una calibración 2D, esto es, permite proyectar puntos de la imagen sobre un plano 2D definido del espacio objeto. La nube de gotas a escanear es 3D, sin embargo, en cada corte todas las gotas capturadas pasa por un plano definido por el sistema de iluminación. Luego la calibración 2D se puede utilizar en este Proyecto si se obtienen los parámetros de la cámara para cada corte.

Se podría pensar en realizar una calibración 3D, de modo que se ahorraría tener que hacer una calibración de la cámara para cada uno de los planos definidos por las paradas del sistema de escaneado. No obstante, este tipo de calibración se muestra más compleja (ActiveMIL no provee la funcionalidad suficiente para llevarla a cabo), además de resultar más imprecisa.

5.5.2. Controles ActiveMIL utilizados

Para realizar las diferentes calibraciones de la cámara, el software de este Proyecto se ha servido del control ActiveX que ActiveMIL provee para tales efectos.

El control para calibración de ActiveMIL consiste en un conjunto de métodos, propiedades y eventos que permiten la realización de una calibración 2D de la cámara, permitiendo el mapeado de las coordenadas en píxeles de las imágenes sobre las coordenadas del espacio objeto. Entonces, para poder visualizar el corte sobre el plano objeto en una imagen es necesario emplear el método que el control posee para transformar imágenes y que parte de los parámetros en el proceso de la calibración. Una vez la imagen se ha transformado se pueden obtener de ella las medidas de las derivas en centímetros.

Se necesitará un control ActiveX que represente la calibración de la cámara para cada uno de los cortes del sistema de escaneado. Esta opción es válida en este Proyecto porque sólo se usan 4 cortes, si se quisiera expandir este número para no importar demasiados controles habría que usar directamente las funciones que proveen las MIL ó crear unas librerías propias para la calibración.

5.5.3. Proceso de Calibración

La calibración debe realizarse al comenzar la aplicación, ya que sin ella la aplicación no puede generar los resultados.

Empleando ActiveMIL la calibración es muy sencilla desde el punto de vista del software. Una vez se tiene una imagen de la rejilla de puntos de calibración sobre todos los cortes, e información sobre la situación de estos puntos en el espacio de coordenadas de los objetos, se puede obtener los parámetros de calibración de la cámara para ese corte con tan solo una llamada a un método. La información de sobre la situación real de los puntos de la rejilla, se le introduce a cada control fijando coordenadas del punto situado en el extremo superior izquierdo de la rejilla, y la distancia entre filas y columnas en las unidades en que se quieren los resultados.

La aplicación durante la fase de calibración abrirá las imágenes previamente obtenidas para la calibración (ver en apartado 7.3.3 como éstas han de generarse), una para cada corte, y partir de cada una de ellas se calibra el control correspondiente. ActiveMIL se encarga de buscar las coordenadas en la imagen de los puntos de la rejilla. Luego mediante algoritmos de optimización, intentará hacer corresponder con el menor error posible las coordenadas de la rejilla en la imagen, con las coordenadas en el espacio objeto introducidas. La convergencia de estos algoritmos, y por tanto del proceso de calibración, no siempre esta garantizada, por esto en el apartado 7.3.3. se dictan una serie de normas que tratan de asegurar una toma de las imágenes de la rejilla que garantice la convergencia de los algoritmos de calibración.

5.6. Obtención de Resultados

Una vez que la información interesante de las imágenes se ha aislado de la mayoría del ruido, mediante el preprocesado que se realiza durante la adquisición de imágenes en el proceso de escaneado, se dispone de información suficiente para poder transformarla desde el sistema de coordenadas de la imagen (en píxeles) al sistema de coordenadas de los objetos (en centímetros), entonces ya se puede comenzar con el análisis para la obtención de los resultados de las imágenes. Esta fase de la aplicación se corresponde principalmente con las etapas 4 – 7 del modelo de sistema de visión por computador presentado en el capítulo 2.

La aplicación finalmente de cada corte presentará dos vistas; una en la que se puede ver toda el área del corte en perspectiva (sistema de coordenadas de la cámara) con el valor de las derivas en centímetros, y otra en la que se tiene una vista cercana y en el sistema de coordenadas objetos de una estimación del flujo promedio de líquido en el corte.

5.6.1. Análisis del Problema

Se parte del resultado obtenido en la preprocesado de imágenes, esto es, una imagen con todas las gotas capturadas durante la fase de adquisición de imágenes, en la que la mayoría del ruido se ha eliminado. A veces puede que aún permanezca sobre la imagen objetos fuera de la zona del corte (restos de ruido, reflejos,...), que en la etapa de análisis pudiera distorsionar las medidas, es por esto que se necesita una nueva etapa de procesado de imagen, pero esta vez a un nivel mayor (a nivel de objetos), que requiere un análisis previo de las características de cada objeto para determinar cual de ellos es el corte sobre la nube pulverizada.

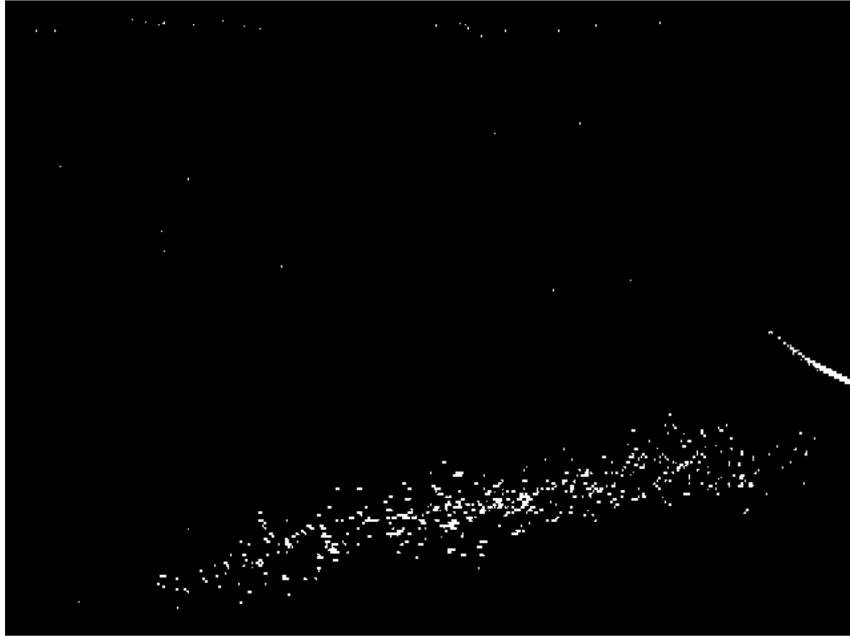


Figura 5.17. Imagen del corte más cercano al suelo, se puede apreciar como existe objetos fuera del corte de la nube, como son pequeños restos de ruido del sensor, y un reflejo del láser en el cajón (en el centro y a la derecha de la imagen). En este corte las derivas son máximas y por tanto el flujo promedio está muy disperso.

La primera vista se limitará a mostrar el área que ocupará la región del corte indicando sobre ella las distancias que se van a tomar como derivas, y sus valores en grados y centímetros. Para ello previamente deben eliminarse los objetos no incluidos en la región. Esta vista es interesante ya que facilita una vista desde el sistema de coordenadas de la cámara y ofrece al usuario tener una percepción gráfica de la definición de deriva, también permite extraer conclusiones cuantitativas ya que muestra los resultados numéricos. Aunque hay que recordar que los ángulos y distancias de estos segmentos, al estar en el sistema de coordenadas de la cámara, no son los reales, estos han de extraerse de los resultados numéricos.



Figura 5.18. Vista de un corte intermedio. Comparándolo con el de la figura anterior este está menos disperso.

Para poder obtener los resultados numéricos correctos es necesario tirar de la información obtenida durante la calibración. Esta información va a permitir transformar la imagen al sistema de referencia de los objetos, luego mediante algoritmos de análisis de imagen se obtienen los resultados numéricos. Estos resultados también se podrían haber obtenido, en la unidad y sistema de coordenadas correctos, si analiza directamente la imagen preprocesada (sin transformarla con la información de calibración) y luego los resultados extraídos se transforman utilizando la información de la calibración. Sin embargo, este proceso, aunque más rápido, es menos preciso, y puesto que a este proceso de obtención de resultado no se le exige velocidad no se adoptará esta solución. Hay que decir que a este proceso no se le exige velocidad ya que se ejecutará después de que se haya llevado a cabo el proceso de escaneado, tomando como partida las imágenes que éste genera, luego dispone de toda la paciencia del usuario (unos segundos).

De todas formas, al final es necesario transformar la imagen al sistema de coordenadas de los objetos para poder generar la vista de la distribución de flujo de las gotas de líquido. El resultado de la transformación al sistema de coordenadas objeto, genera una nueva imagen con una vista desde un plano paralelo al del corte, con un tamaño que encaja en la imagen anterior. Debido a esto, es necesario realizar una serie de operaciones geométricas en la imagen para tener una vista centrada del corte y que aproveche todo el tamaño de la imagen.

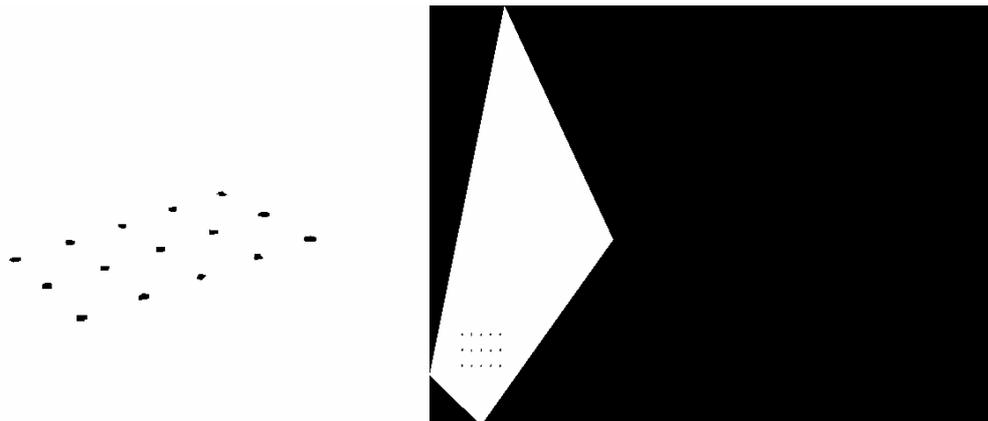


Figura 5. 19. Transformación de la imagen de un grid de puntos rectangulares al sistema de coordenadas de los objetos. La nueva vista se encaja en una imagen de 640x480 píxeles, el grid aparece corregido pero no se aprovecha el tamaño de la imagen.

Para obtener un flujo promedio representativo es necesario realizar un nuevo procesamiento sobre la imagen calibrada. Se tiene una imagen que posee la suma de los destellos generados por las gotas que durante un tiempo determinado atraviesan un haz de luz plano. Esto se quiere pasar una especie de distribución de probabilidad 2D, en sentido que dado un corte de una idea cualitativa, en forma de intensidad de píxeles, de la probabilidad que una gota aislada pase por una determinada zona de ese corte.

Esta distribución de probabilidad aproximará la distribución de flujo relativo a ese corte, es decir, dado el flujo de gotas que atraviesa el corte, habla de cómo éstas se distribuyen sobre él. No se va a estimar el flujo absoluto ya que el modelo de sistema de escaneado presentado permite contar las gotas pero no medir su tamaño, luego esta estimación puede que no tenga la precisión suficiente para realizar medidas cuantitativas. En resumen, se va a tener para cada corte una idea aproximada de la distribución del flujo relativa a la cantidad de flujo que atraviesa ese corte, lo

que va ha permitir hacer estudios cualitativos de la homogeneidad con la que riega el pulverizador.

5.6.2. Controles ActiveMIL utilizados

Para realizar las operaciones de este proceso además de otros controles ActiveX de ActiveMIL ya descritos en otros apartados, para el manejo de imágenes, procesado de imágenes y calibración, se han utilizado dos nuevos, uno para el análisis de imágenes y otro para dibujar y escribir sobre ellas.

Para el análisis de imágenes se utiliza el control llamado *BlobAnalysis*, el cual permite extraer características de regiones conectadas (comúnmente conocidas como *blobs*) de una imagen. Este control funciona sobre imágenes en una banda de 1, 8 ó 16 bits, por lo que se pueden usar las imágenes de 1 banda y 8 bits empleadas hasta ahora. Puede trabajar en imágenes en escala de grises, lo cual relentiza bastante sus algoritmos, no obstante, esto no ha sido problema en esta aplicación ya siempre se han analizado imágenes binarias, píxeles a 0 (nivel mínimo) o 255 (nivel máximo en imagen de 8 bit).

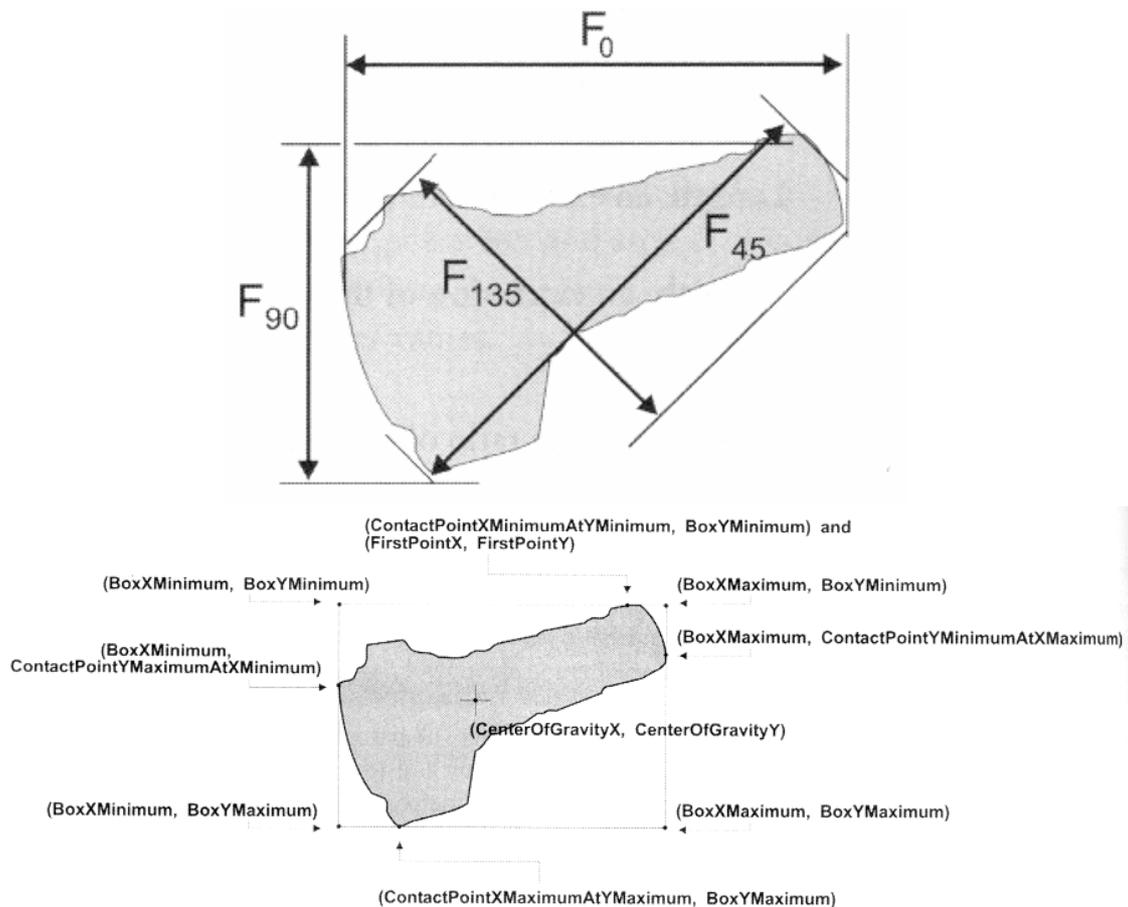


Figura 5. 20. Definición de parámetros box y diámetros de Feret.

A través de este control se puede extraer gran cantidad de características de los blobs, de entre las que se encuentran las necesarias para esta aplicación, como son los parámetros *box* y los diámetros de Feret, área. Los primeros se definen como los parámetros que describen el mínimo rectángulo que circunscribe al blob y cuyos lados son paralelos a los de la imagen, estos

parámetros se emplearán en las operaciones geométricas de rotación y zoom para mejorar la presentación de la vista de flujo. Los diámetros de Feret máximo y mínimo definirán las derivas máximas y mínimas, que una vez corregida la imagen con el control de calibración, proporcionarán el resultado en centímetros. La precisión en la obtención de los diámetros de Feret máximo y mínimo dependerá de la cantidad de diámetros de Feret se obtengan, en esta aplicación se obtienen 16. El área de los blobs se utiliza para el filtrado que se realiza en la fase de preprocesado a alto nivel (ver apartado 5.6.3).

El control empleado para dibujar y escribir sobre las imágenes se llama *GraphicContext*. Gracias a él se ha escrito el texto en el que se informa sobre el tamaño y ángulo de las derivas, además de dibujar los segmentos que las muestran sobre la vista de las derivas. Aunque en este Proyecto no se aprovecha también provee funcionalidad para dibujar puntos, cruces, rectángulos y arcos.

5.6.3. Etapa de Procesado a Alto Nivel

Se dice que esta etapa es de procesado a alto nivel porque para modificar la imagen necesita un análisis previo de las características de los objetos, eliminando lo que según estas características se estima que no pertenecen al corte de la nube.

En principio las imágenes obtenidas en el proceso de escaneado poseen multitud de blobs, en su mayoría son resultado de la captura e integración de destellos de luz reflejados por las gotas de la nube, estos estarán concentrados en la región de la imagen que corresponde a donde la luz corta a la nube y a la que quiere medir sus longitudes máxima y mínima (derivadas). Su concentración será mayor o menor dependiendo la altura la que se realiza el corte, pudiendo ser tan alta (cortes cercanos al pulverizador) que estos pequeños blobs se superpongan entre sí definiendo uno o unos pocos grandes blobs (ver imágenes 5.17 y 5.18). Fuera de esta región, que es la interesante, pueden aparecer algunos puntos (pequeños blobs) “expúreos” que provengan del ruido del sensor que en la etapa de preprocesado inicial no se hayan conseguido eliminar, o incluso blobs que provengan de reflejos de la luz del láser sobre el cajón.

Para poder definir la región se divide la imagen en casillas de tamaño 16x16, en cada casilla se suma el total de píxeles a nivel alto (hay que recordar que se trabaja con imágenes que previamente se han binarizado), y se ponen todos los píxeles de la casilla al nivel que indica la suma, después se binariza según umbral elegido gracias a la experiencia. Este proceso permite hallar cuales son las regiones de la imagen obtenida en el escaneado, que aún se almacena, en forma de grandes blobs donde existe una alta probabilidad de que un píxel este a nivel alto.

$$Nivel_{16 \times 16} = \sum_{j=y_0}^{y_0+16} \sum_{i=x_0}^{x_0+16} b(i, j) \quad b(x, y) \in (0,1) \quad (5.6)$$

La elección de celdas de tamaño 16x16 píxeles responde a tres requisitos, es cuadrada, sus dos lados son divisores tanto del ancho como del largo de las imágenes luego permite definir un número entero de celdas, y es un tamaño justo para que tras el binarizado condensar en un solo blob los pequeños blobs que definen cada uno de los objetos de la imagen, ya sea la región de corte de la fuente de iluminación sobre la nube de puntos o un reflejo.

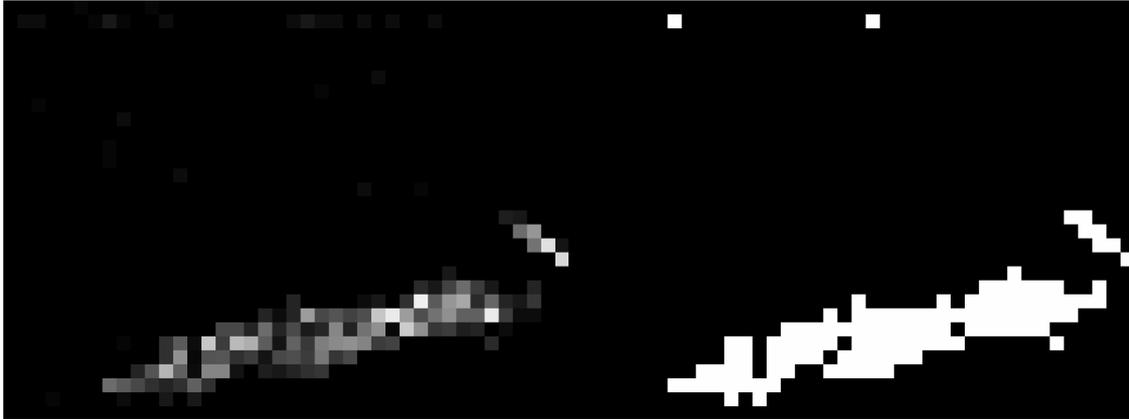


Figura 5.21. Promediado para la definición de regiones de la imagen de la figura 5.17 (izquierda), tras la binarización (derecha) el ruido que no se ha eliminado se muestra como blobs del tamaño de una celda, de los dos grandes blobs, uno representa al corte (centro) y el otro a un reflejo de luz sobre el cajón (derecha).

Un blob que represente la región del corte presenta una serie de características que permiten aislarlo del resto de blobs. Las que se han utilizado en esta aplicación son: debe de tener un área mínima, y que debido a la colocación de la cámara no puede estar en el extremo superior de la imagen, ni muy escorado hacia los laterales de la imagen.

El área se ha escogido mínima con un valor de 512, que equivale a la superficie de dos de las celdas. Con esta área mínima se filtran los pequeños puntos provenientes del ruido de sensor, ya que estos son pocos y muy repartidos por la imagen, luego tras el promediado y binarizado muy raro sería que formasen blobs mayores de dos celdas. Con la información sobre la posición se filtran los blobs correspondientes a reflejos de luz, debido a que estos suelen estar localizados en los extremos de la imagen debido a que se suelen generar en las paredes del cajón.

Para determinar la posición de los blobs se ha utilizado los parámetros box explicados en el apartado anterior. Para calcular el área se parte de su definición para superficies bidimensionales:

$$\text{Area} = \iint f(x, y) dx dy \quad (5.6)$$

Pero particularizando para el caso de superficies discretas (x e y , coordenadas en la imagen solo toman valores discretos) formadas por píxeles, luego:

$$f(x, y) = b(x, y) = \begin{cases} 1 & \forall (x, y) \in \text{objeto} \\ 0 & \text{resto (fondo)} \end{cases} \quad (5.7)$$

Según estas expresiones el control analizador de blobs puede calcular las áreas de cada uno, ya que puede aislarlos y para cada uno sumar todos los píxeles que lo componen.

Una vez se queda solo el blob que define la región del corte sobre la imagen, se multiplica ésta con la imagen que se tenía almacenada resultado del proceso de escaneado. La multiplicación en imágenes es píxel a píxel (nada tiene que ver con la multiplicación entre matrices), como ambas son imágenes binarias, el resultado es la operación lógica AND entre ellas, de manera que en la imagen resultado del escaneado solo queda la información contenida en el área que se ha definida como región de corte. Dicho de otro modo, se elimina toda la información que no interesa y que el preprocesado realizado durante la fase de escaneado no había podido eliminar.

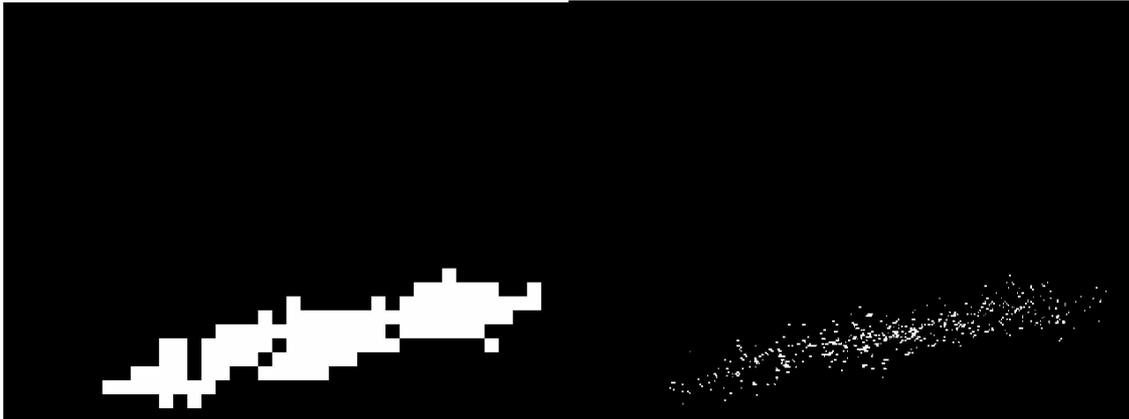


Figura 5.22. (Izquierda) Máscara resultado del filtrado de la imagen binarizada de la figura 5.21, la cual se emplea para a través de un AND con la imagen de la figura 5.17 obtener una imagen con solo la información interesante (derecha).

Esta etapa de procesado a alto nivel para cada corte se resume en los siguientes pasos:

1. Copiar la imagen obtenida en la fase de escaneado (imagen1)
2. Dividir la imagen copiada (imagen2) en celdas 16x16 píxeles
3. Contar el número de píxeles activos de cada celda, y establecer el nivel de cada celda en imagen2 a ese valor
4. Umbralizar la imagen2
5. Extraer los blobs de imagen2
6. Calcular el área y posición de cada blob
7. Eliminar blob si ($\text{area} < 512 \mid \text{Ymin} < 40 \mid \text{Xmin} > 400$)
8. Imagen resultado = imagen1 AND imagen2

Figura 5.23. Pasos principales de la etapa de procesado a alto nivel.

5.6.4. Obtención de los Resultados

Una vez se dispone de una imagen con solo la información relevante, obtenida tras la etapa de procesado a alto nivel, se pasa a extraer los resultados y generar las vistas finales para el usuario.

En primer lugar se transforma, o calibra, la imagen al sistema de coordenadas de los objetos, para así poder extraer el valor correcto de los parámetros. Se extraen los diámetros de Feret máximo y mínimo en centímetros, además de sus ángulos, estos son resultados numéricos de las derivas. Luego sobre la imagen que se utilizó como máscara para delimitar la región que ocupa la información del corte de la nube, se escriben los resultados numéricos y se dibuja los segmentos que definen los ángulos de Feret. Esta vista además de mostrar los resultados numéricos, da una idea sobre la región que ocupa el corte desde la perspectiva de la cámara y de la definición de derivas.

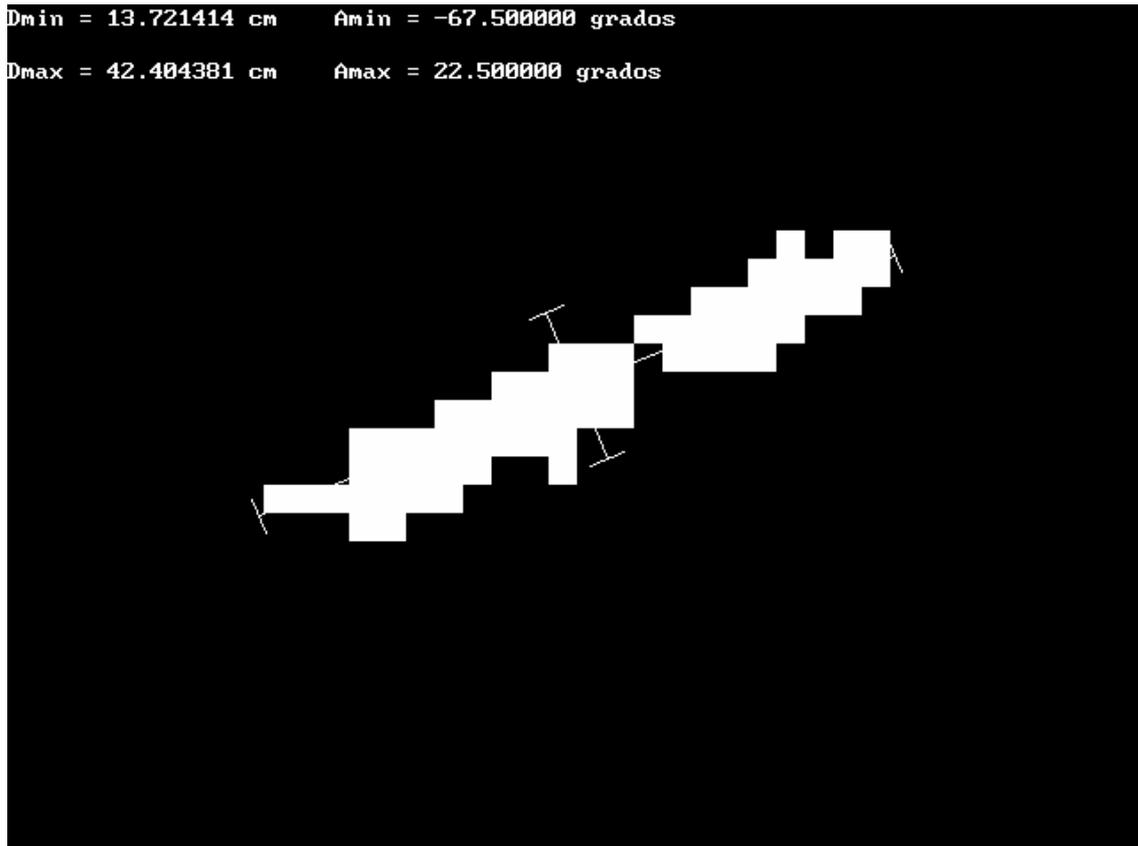


Figura 5. 24. Vista con la presentación de resultados numéricos y definición de derivas.

Después se procede a generar la vista del flujo de gotas de líquido. Sobre la imagen calibrada se analizan los parámetros box y ángulo de Feret del diámetro máximo esta vez en unidades de píxeles. Esta información se utiliza para realizar el zoom y las rotaciones de la imagen necesarias, para conseguir tener una imagen con la información del corte centrada y aprovechando toda el área de la imagen.



Figura 5. 25. (Izquierda) Imagen de un corte transformada con la información de calibración. (Derecha) Zoom sobre la información del corte.

Para tener una estimación de la densidad de flujo de gotas relativo, pseudo-densidad de probabilidad de que una gota escogida al azar atravesase el corte por esa zona, se realiza un

filtrado paso bajo aplicando el filtro de respuesta al impulso infinita (IIR) de Canny-Deriche. Este filtro contiene una función de ponderación exponencial de la forma:

$$K(\alpha|n| + 1)e^{-\alpha|n|} \quad (5.8) [8]$$

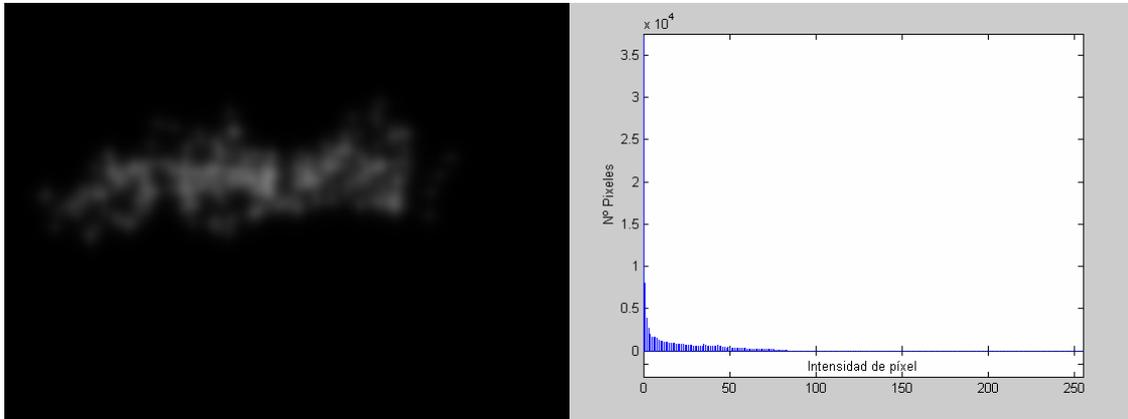


Figura 5. 26. Imagen resultado del filtrado por Canny-Deriche y su histograma.

La imagen obtenida tras el filtrado posee un bajo contraste (ver histograma de la figura 5.26), para mejorarlo se realiza una ecualización de histograma. El histograma de la imagen filtrada sigue una distribución exponencial, luego se ecualiza el histograma empleando un modelo de densidad de probabilidad de salida ($P_g(g)$) también exponencial.

$$P_g(g) = \alpha(e^{-(\alpha)(g-g_{\min})}) \quad g \geq g_{\min} \quad (5.9) [8]$$

Representado g cada una de las intensidades de la imagen (*bins* del histograma), y α un parámetro que define la forma de la probabilidad y que en esta aplicación se ha elegido a partir de la experiencia. $P_g(g)$ define la función de transferencia del histograma:

$$g = g_{\min} - \frac{1}{\alpha} \ln(1 - P_f(f)) \quad (5.10) [8]$$

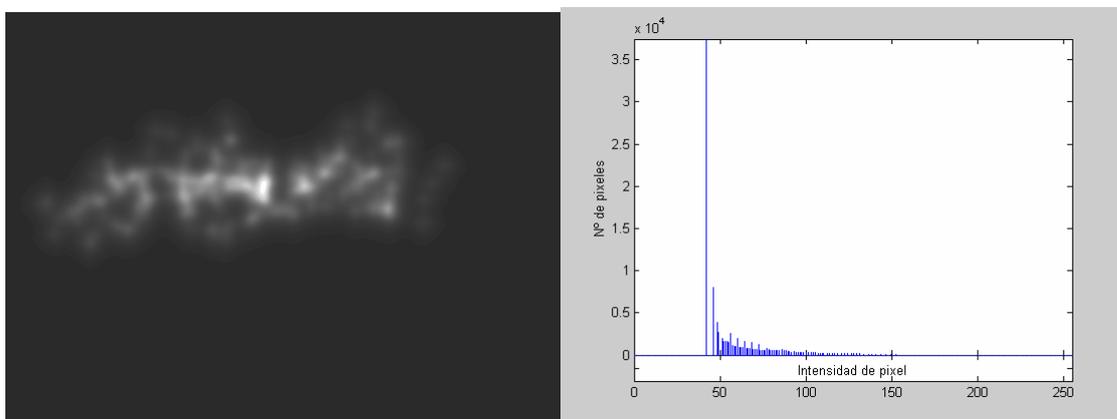


Figura 5. 27. Imagen con contraste realizado tras la ecualización del histograma de la imagen de la figura 5.26 y su histograma.

La distribución de probabilidad acumulada, $P_f(f)$, de la imagen es aproximadamente la acumulación del histograma:

$$P_f(f) \approx \sum_{m=0}^i H_F(m) \quad (5.11) [8]$$

El ajuste del histograma con este método no es perfecto, pero esto se ha hecho a propósito para tener una mayor luminosidad en la imagen y región del corte esté más resaltada del fondo. Para facilitar la distinción entre el fondo y la región de interés, los bits del fondo son puestos a cero, a partir del histograma se puede conocer cuales son estos bits (en el histograma de la figura 5.27 se representan como los de menor intensidad).

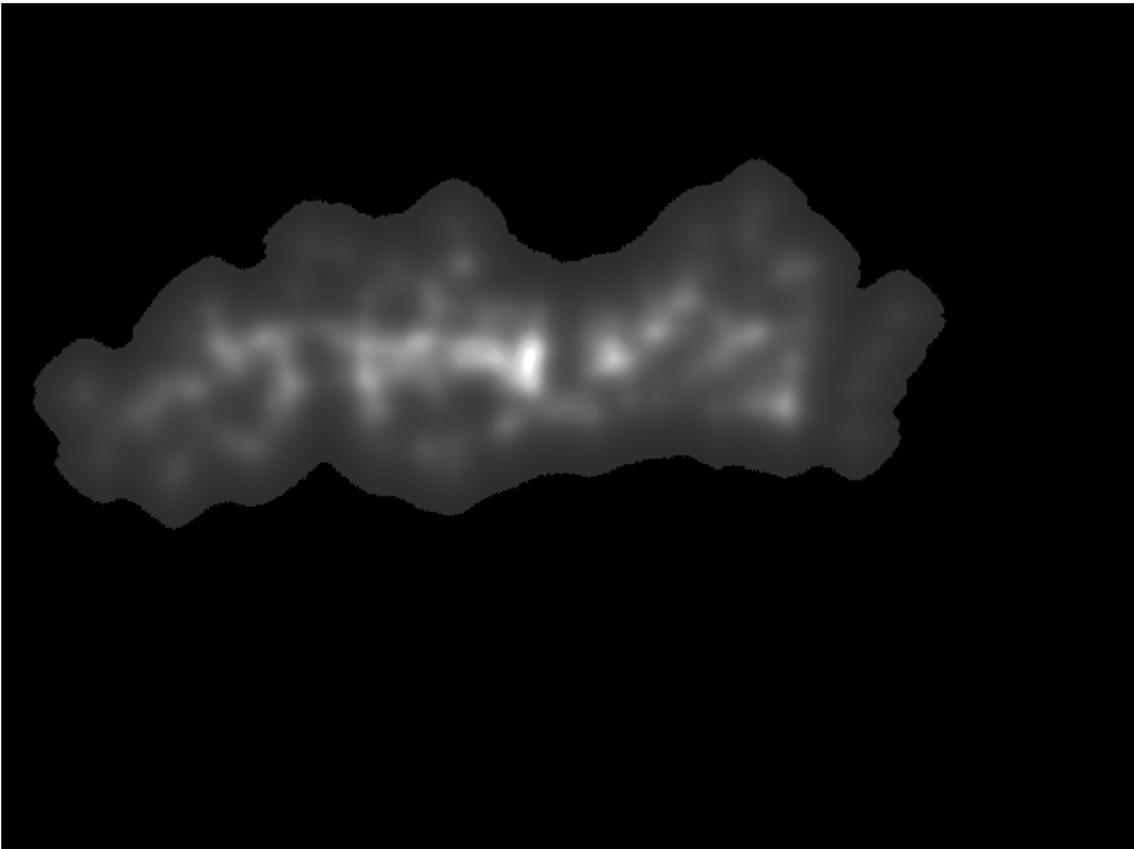


Figura 5. 28. Vista de la distribución de flujo promedio de gotas de líquido y aspecto de la región de un corte.

El proceso de obtención de los valores de las derivas y generación de la vista de distribución de flujo se expone a continuación:

1. Transformar (calibrar) la imagen con la información de interés (imagen1)
2. Binarizar imagen1
3. Calcular los diámetros (en cm y píxeles) y ángulo de Feret máximos y mínimo (derivas valores en cm)
3. Calcular el centro de la nube de blobs de imagen1
4. Rotar con ángulo de Feret máximo imagen1, tomando como centro el centro de la nube de blobs
5. Hacer zoom sobre el centro de la nube de blobs, y según el

ancho que define el diámetro máximo de Feret en píxeles

6. Aplicar filtro Canny-Deriche
8. Ecualizar histograma
7. Poner bits de fondo a cero

Figura 5. 29. Resumen de los pasos principales del proceso de obtención de resultados y generación de la vista de flujo.

CAPÍTULO 6

IMPLEMENTACION DEL SOFTWARE DE CONTROL

En este capítulo se describirá la lógica de la aplicación desarrollada como parte del sistema de escaneado, controlado por una autómatas, desarrollado en este Proyecto. En primer lugar, se realizará un breve resumen de las herramientas empleadas para su implementación. Después se pasará a describir la aplicación desarrollada, haciendo un análisis del problema, describiendo como se realiza la comunicación con el PC que contiene la aplicación de visión y explicando su funcionamiento.

6.1. Herramientas empleadas para el desarrollo del software de control

La elección del entorno en el que se ha desarrollado la aplicación de control, viene condicionada por la elección del autómatas en que se va a ejecutar. Esto se debe a que el fabricante del autómatas es también el proveedor del software que permite su configuración y programación.

La elección del lenguaje de programación a utilizar en este caso viene limitada por las herramientas empleadas. Aún así, la herramienta de programación del autómatas permitía la programación en varios lenguajes con propiedades distintas, atendiendo ahora el criterio de elección a su sencillez en el manejo y la facilidad en la depuración.

6.1.1. Entorno de Programación

Junto con el autómatas de Siemens® empleado en este Proyecto, se disponía de software de este mismo fabricante para realizar las tareas de programación y configuración del autómatas. Para la programación en concreto se ha utilizado la herramienta que provee el *Administrador Simatic (Simatic Manager)*, instalado a través del CD SIMATIC NET 6.2+SP1.

Este CD además de la herramienta de programación, contiene los drivers del módulo de comunicación PROFIBUS instalado en el PC, un servidor OPC, herramientas de configuración y diagnóstico de enlaces industriales, etc. Este software ha sido instalado en el PC donde se ejecuta la aplicación de visión, de manera que a través del módulo PROFIBUS que tiene instalado y el servidor OPC que provee el software se realiza la comunicación con el autómat. Se ha aprovechado que se debía instalar el software de Siemens® para realizar las tareas de comunicación, para configurar y programar el autómat desde el PC donde se ejecuta la aplicación de visión artificial gracias a las herramientas que integra el Administrador Simatic.

El Administrador Simatic integra varias herramientas que permiten la programación, configuración, y diagnósticos de autómat Siemens®. En concreto la herramienta de programación permite programar el autómat empleando varios lenguajes, una vez escrito el programa se compila en el PC y se envía a la memoria del autómat, desde donde puede ser ejecutado, a través del enlace PROFIBUS creado entre PC y autómat, que luego empleará la propia aplicación para comunicarse con el sistema de visión. La herramienta de programación, también a través del enlace PROFIBUS, permite hacer un diagnóstico en línea de la aplicación, en la que mientras la aplicación se ejecuta en el autómat se puede ver el valor de las variables, qué condiciones se cumplen y las partes del código que se están ejecutando en cada momento. Esta funcionalidad ha facilitado en gran medida la depuración del código.

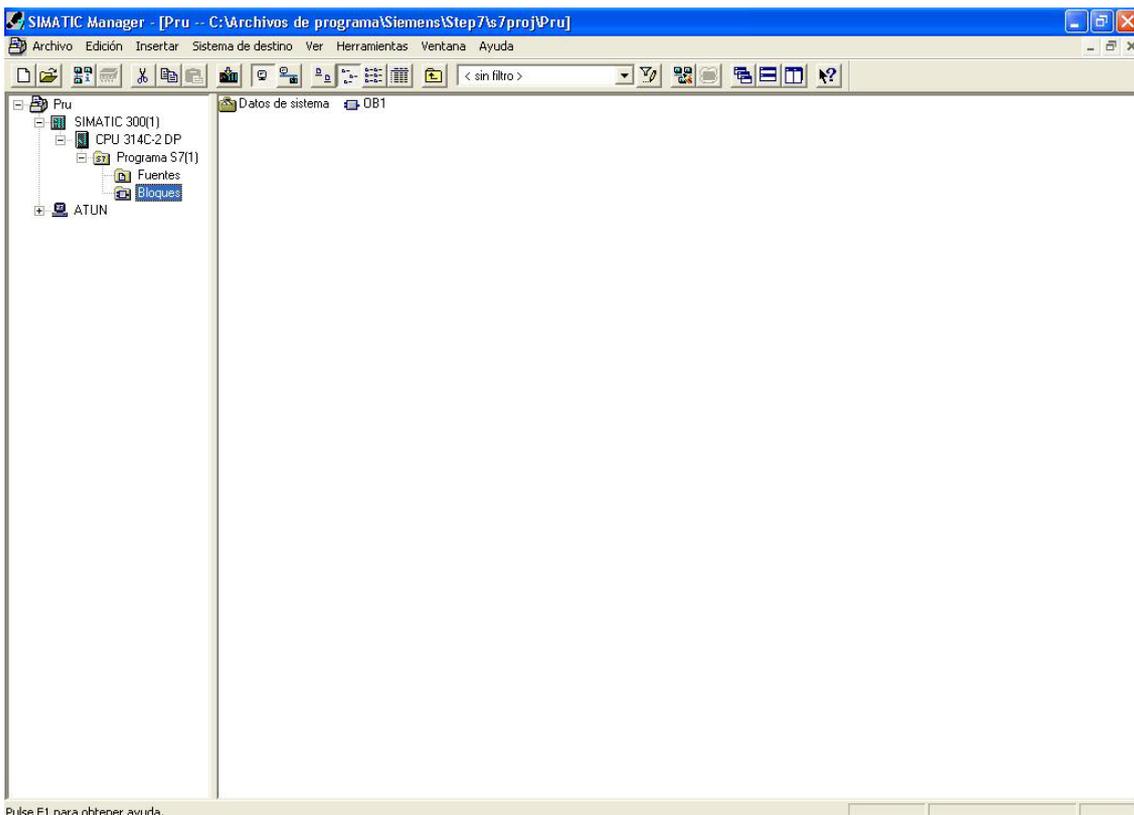


Figura 6. 1. Administrador Simatic, entorno en el que se ha configurado y programado al autómat.

6.1.2. Lenguaje de Programación

La herramienta de programación utilizada ofrece la posibilidad de desarrollar programas para autómatas S7 en tres lenguajes; AWL, KOP y FUP. Cada lenguaje tiene unas propiedades distintas, lo que confiere a cada uno unas ventajas e inconvenientes sobre los otros.

En este Proyecto se ha empleado KOP debido a lo sencillo de su aprendizaje, factor importante porque no se tenía experiencia previa en la programación de autómatas antes de realizar el Proyecto, y a la facilidad con que se puede depurar el código si se combina con el diagnóstico en línea que permite la herramienta de programación. KOP permite estructurar fácilmente los programas debido a que se trabaja con cajas que definen operaciones o funciones, y que se pueden conectar entre sí o con las entradas o salidas del autómata. De esta forma es muy fácil ver que pasos sigue el programa según el estado de cada variable.

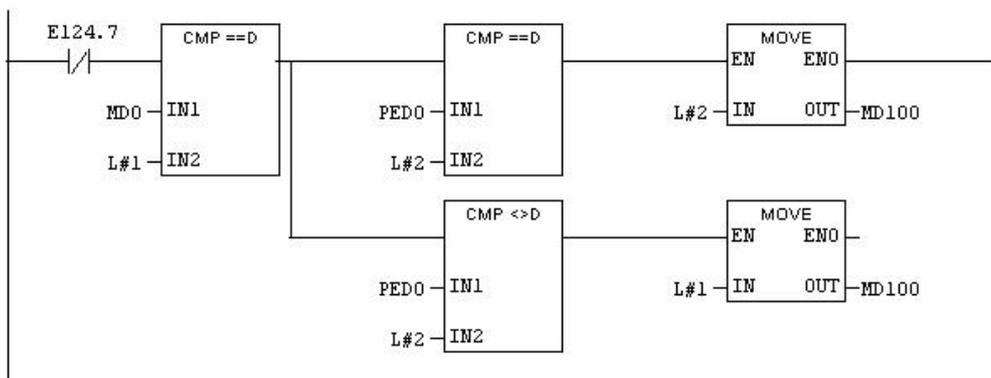


Figura 6. 2. Aspecto de un segmento de código en lenguaje KOP.

El problema KOP es que sus cajas necesitan una sistemática de por parte de la herramienta de programación que hace que no se optimice el código en las mismas, al menos en comparación con lo que se podría optimizar con un lenguaje basado en texto como es AWL [14]. Sin embargo, los requisitos de la aplicación y la potencia del autómata empleado (CPU 314C-2DP), hacen que esta menor eficiencia en el código no sea un inconveniente. FUP es un lenguaje que también trabaja con cajas y con propiedades parecidas a KOP, aunque no expresa de manera tan evidente las combinaciones AND y OR, a como lo hace KOP.

6.2. La Aplicación de Control

Sobre el autómata se ejecuta una aplicación que se encarga de leer y escribir respectivamente en las entradas y salidas del sistema de escaneado, para así controlarlo. Además, gracias a una colaboración con la aplicación de visión a través de PROFIBUS el proceso de escaneado se realiza de forma satisfactoria.

6.2.1. Análisis del Problema

A las entradas del autómata se conectan los sensores del sistema de escaneado, y dos de sus salidas a la caja de electroválvulas que controla el actuador neumático, según el esquema eléctrico del apartado 4.7.5. En el extremo del vástago del actuador neumático se sitúa la fuente de iluminación del sistema de escaneado, que debe ser movida a lo largo del recorrido del

actuador y parándose a la altura que se defina para cada corte que se va a realizar a la nube de líquido.

Los dos sensores de proximidad van a definir el recorrido del actuador, y a la vez el corte superior e inferior. Cuando el actuador llega a uno de estos extremos el sensor correspondiente se activa, generando un flanco de tensión de 24V que será leído por la entrada digital correspondiente.

El sensor de ultrasonidos genera una señal de corriente analógica, la cual varía linealmente en función de la altura a la que éste se sitúa. Esta señal se recoge en una entrada analógica de corriente, que digitaliza su valor, lo que permite manejarlo desde la CPU del autómat. Luego para conocer la altura en cada momento a la que se encuentra el sensor, y así poder decidir cuando parar para realizar un corte, basta con realizar una simple transformación lineal (ver figura 6.3). La pendiente de esta curva depende del rango que se haya definido para el sensor (ver apartado 7.4.2).

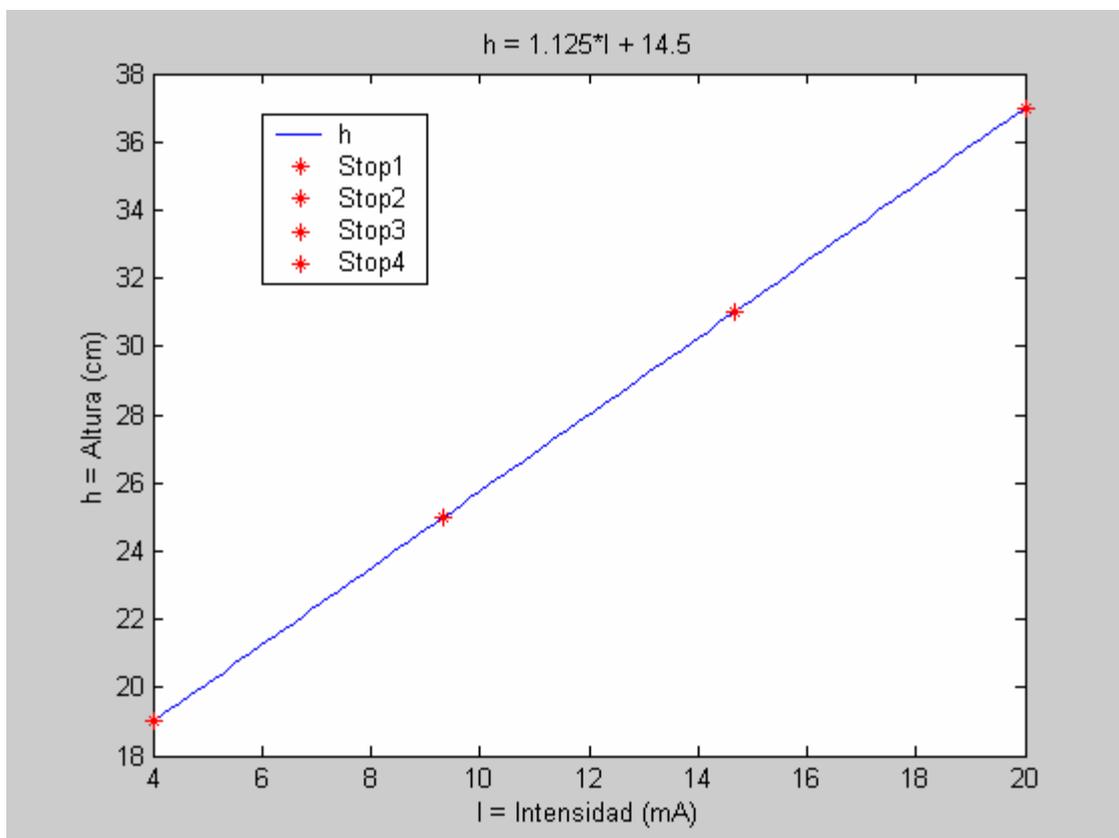


Figura 6. 3. Función lineal que permite calcular la altura a la que se encuentra el sensor de ultrasonidos en función de la intensidad que genera, según los límites (19-37cm) empleados en este Proyecto. Las marcas en rojo indican cuando se realizarán las paradas (stop).

Los cortes se eligen de tal manera que la distancia entre ellos sea la misma en todos los casos, esto es, para cuando la altura del sensor es de 19, 25, 31 y 37 cm del suelo (distancia de 6cm en todos los casos), que se corresponde con alturas de los cortes (el laser está 1 cm por debajo del sensor) de 18, 24, 30 y 36 cm. Si las paradas inferior (19cm) y superior (37cm) están ya definidas por los sensores de proximidad, la aplicación de control sólo debe esperar a leer por su

entrada analógica 9.33 y 14.66 mA que son los valores de corriente que se corresponden con las dos paradas intermedias.

El autómatas dispone de un circuito digitalizador de 11bits, que automáticamente muestrea y cuantifica la señal de entrada analógica, y la va almacenando en un registro de la memoria del autómatas. El programa de control leerá el registro según lo especifique el formato de tipo de datos que esté utilizando, habiendo de ser capaz de interpretar cuando uno de estos valores se corresponde con la altura correspondiente a una parada.

	Parada 1	Parada 2	Parada 3	Parada 4
Altura (cm)	19	25	31	37
S. Analógica (mA)	4	9.33	14.66	20
Registro (11bit)	0	683	1365	2048
WORD (16bit)	0	10917	21835	32752

Tabla 6. 1. Conversión de datos entre la altura real del sensor y los valores que maneja la aplicación de control.

En realidad la aplicación de control debe realizar las paradas antes de los valores indicados en la tabla 6.1. Debido a que ha de compensar el tiempo de digitalización, el tiempo que el programa tarda en determinar que se ha llegado a una parada, pero sobre todo el tiempo que tarda en estabilizarse el actuador neumático. Los valores de la tabla 6.1 son sólo cotas máximas, estos valores en realidad se han ajustado a partir de la experiencia.

Los valores de la tabla 6.1 se refieren a la configuración empleada en este Proyecto, si por algún motivo se quisiera cambiar el rango en el que se moverá el actuador, cambiando la posición de los sensores de proximidad, por ejemplo, si desea estudiar un sensor colocado a una altura distinta, sólo sería necesario recalibrar el sensor de ultrasonidos (ver apartado 7.4.2) para que su salida se ajuste al nuevo rango de altitud. Puesto que el rango de intensidad de salida del sensor es fijo, las paradas seguirían siendo equidistantes.

Cuando arranca un proceso de escaneado y el tiempo que se a de permanecer en una parada determinada lo decide la aplicación de visión, que se lo comunica a la aplicación de control a través del enlace PROFIBUS-DP y el servidor OPC. Además, para el correcto funcionamiento de la aplicación de visión, ésta ha de saber cuando se ha de capturar y el corte que se está capturando, por esto la aplicación de control ha de notificarle en todo momento cual es su estado actual.

Para facilitar el desarrollo de la propia aplicación de control y del sistema en conjunto, la aplicación ha de tener un *modo manual*. Es decir, activando las entradas digitales del autómatas mediante interruptores, se ha de poder controlar manualmente el movimiento del actuador neumático. Este modo no sólo ha sido útil durante la fase de desarrollo del Proyecto, para hacer cortes antes de tener el enlace PROFIBUS-DP, sino que una vez implementado el sistema, este modo de ejecución se debe utilizar para llevar al actuador a su posición inicial cuando ha ocurrido un fallo en el sistema de escaneado, y el actuador no terminado en esta posición. También es útil para ajustar el rango de escaneado.

6.2.2. Comunicación con la Aplicación de Visión y los Periféricos

Según la configuración realizada, el autómatas actúa como maestro DP sobre el bus PROFIBUS, al que el PC, donde se ejecuta la aplicación de visión, se conecta como esclavo. Según esta

configuración, el autómata va a disponer de una serie de direcciones de memoria a las que puede acceder el PC a través de bus. Estas direcciones se conocen como de periferia distribuida (DP) y pueden ser de entrada (PE) o salida (PA). La periferia distribuida puede alcanzar un tamaño de 122bytes de entrada y 122bytes de salidas, sin embargo en la implementación de este sistema solo se ha empleado un byte de entrada para recibir desde la aplicación de visión los órdenes codificados como números enteros, y otro byte de salida informar mediante números enteros del estado en que se encuentra la aplicación de control.

Las entradas y salidas, tanto analógicas como digitales, también se encuentran mapeadas sobre direcciones de la memoria del autómata. Gracias a esto, escribir en un actuador o leer desde un sensor, al igual que en el caso de la comunicación con la aplicación de visión, para la aplicación de control se convierte simplemente en escribir o leer sobre uno o varios bits de la memoria. Entradas/Salidas digitales se codifican mediante un bit, y las analógicas mediante 11bits.

DIRECCIÓN		Propósito
Periferia Distribuida	E1023	Recepción de órdenes desde el PC
	S1023	Envío de estado al PC
Entrada Digital	E124.0	Detección activación sensor de proximidad superior
	E124.1	Detección activación sensor de proximidad inferior
Salida Digital	S124.0	Activar electroválvula para bajar
	S124.1	Activar electroválvula para subir
Entrada Analógica	E754	Leer valor de intensidad recibida del sensor de ultrasonidos

Tabla 6. 2. Direcciones de memoria utilizadas por la aplicación mapeadas con dispositivos externos conectados al autómata. E referencia dirección de entrada y S de salida.

6.2.3. Descripción de la Aplicación de Control

La aplicación de control según el análisis del apartado 6.2.1, consistirá en un proceso con una serie de pasos, que generarán unas determinadas salidas, por los que se va moviendo en función de las entradas. Este comportamiento se ajusta al modelo de máquina de estados finitos (FSM), es por esto que la aplicación de control se ha desarrollada siguiendo un esquema de FSM.

ESTADO	ACCIÓN	
CAP_DOWN	Parada inferior (Reset)	
UP	Subir fuente de iluminación	
CAP	Paradas intermedias	
CAP_UP	Parada superior	
DOWN	Bajar a posición inicial	
STOP	Proceso de escaneado terminado	

Tabla 6. 3. Estados de la FSM y diagrama de estado con sus salidas correspondientes según los códigos de la tabla 6.2.

Las entradas y salidas que se emplearán son las mismas que se describen en la tabla 6.2, además de una variable contador (CONT en la tabla 6.3) que el modelo tomará como entrada o salida según el caso, y que le sirve conocer el número de iteraciones en el bucle que se forma

entre los estados UP y CAP (ver figura 6.4). La creación de este bucle y la variable CONT no parece estar muy justificada dado que no ahorra estados para el caso en que se realizan 4 paradas, no obstante ofrece la oportunidad de aumentar el número de paradas con solo incrementar el valor de la variable CONT y la parte del módulo de lógica del estado siguiente que escucha al sensor de ultrasonidos y al PC, sin tener que aumentar el número de estados.

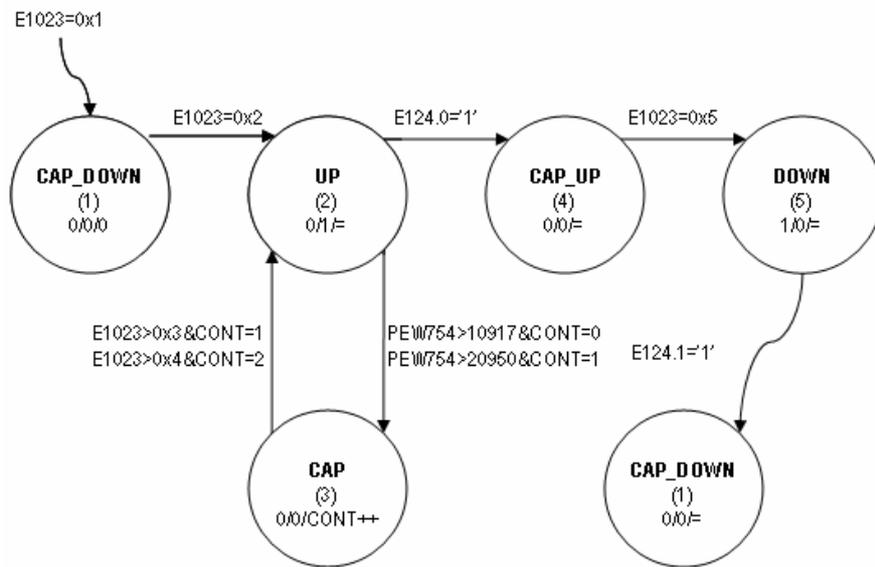


Figura 6. 4. Máquina de estados de la aplicación de control empleando el diagrama de estado de tabla 6.3.

A tenor del modelo FSM presentado en la figura 6.4, se puede concretar y decir que se sigue un modelo de *Máquina de Moore*, puesto que el valor de las salidas solo depende del estado en que se esté, aunque el estado si depende del valor que hayan ido tomando las entradas. Hay que resaltar que un bloque de programa en el autómata utilizado (toda la aplicación se ha programado en un bloque), el código se ejecuta secuencialmente dentro de un ciclo, volviendo a comenzar de nuevo en el siguiente ciclo, con un número de ciclos indefinido. Teniendo esto en cuenta, el software se puede programar siguiendo una arquitectura que aproxime a una arquitectura hardware. Máquina de Moore mezcla de lógica combinacional (operaciones de lógicas, aritméticas,...) y de una lógica secuencial (registros), en la que los pulsos de reloj pueden ser identificados como los ciclos de programa. Luego, esta aplicación tiene una arquitectura software como la descrita en la figura 6.5.

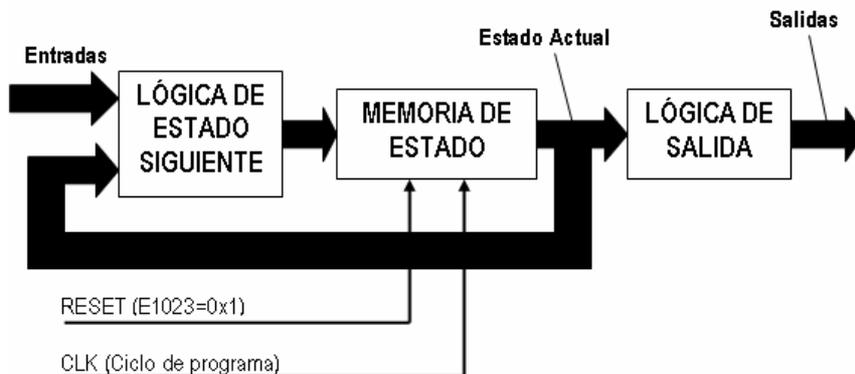


Figura 6. 5. Arquitectura del software de la aplicación.

CAPÍTULO 7

MANUAL DE USUARIO DEL SISTEMA

En este capítulo se describirán los pasos a seguir para la instalación del sistema. También se proporciona una guía sobre la ejecución y manejo de las aplicaciones, y sobre como reajustar los parámetros del sistema. En este capítulo se parte de que el sistema posee todos los componentes que se describen en el capítulo 4, y a su vez están conectados tal y como en este capítulo se especifica.

7.1. Instalación de Software de Visión

En primer lugar se define el software que el PC que va ejecutar la aplicación de visión debe de tener instalado, para que posteriormente se pueda instalar la aplicación. Después se describen los pasos que han de seguirse para una instalación y/o ejecución de la aplicación.

7.1.1. Requisitos Previos

Para la instalación y correcta ejecución de la aplicación de visión artificial en un PC se requiere:

- Utilizar una plataforma PC con el sistema operativo WindowsXP instalado.
- Tener instalado el software del CD MIL 8.0 de Matrox® [8], que contiene los DLLs (*Dinamic Link Library*) para la versión de ActiveMIL utilizada por la aplicación de visión. Pudiera ser que con versiones posteriores también funcionase correctamente, aunque esta posibilidad no ha sido testada. Además, este CD contiene los drivers de la tarjeta de adquisición y cámara IEEE-1394 empleadas en este proyecto, Matrox Meteor-II/1394 y Basler A601fc-2 respectivamente, así como herramientas de configuración del hardware y software de Matrox® que pueden ayudar a mejorar el rendimiento de la aplicación.

- Se podría utilizar otra cámara o tarjeta de adquisición cuyos drivers estuvieran contenidos en el CD MIL 8.0, y siempre estas sean interconectables. Esta posibilidad no se ha testado, y además para el caso de la cámara puede ser que cambie el comportamiento del sistema debido a que posea unas velocidades de captura y sensibilidad distintas a la Basler A601fc-2.
- Disponer de un servidor OPC. El que se ha utilizado en este Proyecto es el que provee Siemens® en el CD SIMATIC NET 6.2+SP1, y en tutoriales y manuales que Siemens® provee en [Web 5] se explica como arrancar el servidor OPC una vez instalado este software.

7.1.2. Instalación y Ejecución

La instalación de la aplicación de visión parte la compilación de un proyecto de Visual C++ .NET, que ha sido el entorno en donde se ha desarrollado la aplicación. No obstante, en el CD del Proyecto solo se provee el archivo ejecutable resultado de esta compilación.

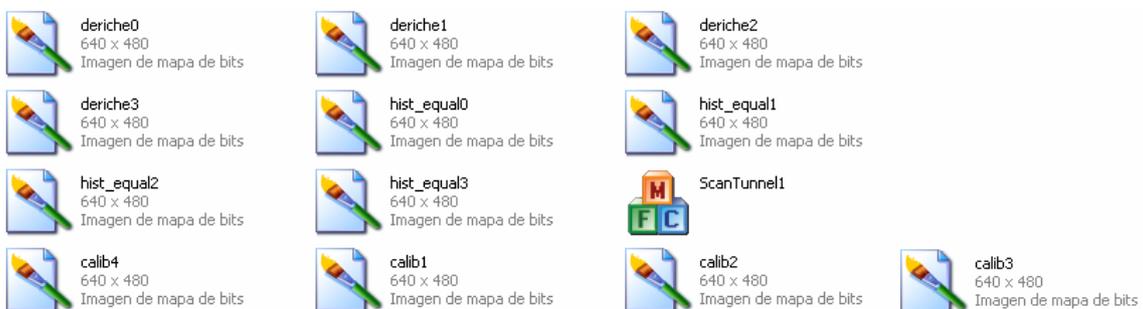


Figura 7. 1. Archivos del directorio donde se instala la aplicación. Debe de aparecer, el ejecutable, las imágenes generadas durante la calibración de la cámara, y si ya se ha ejecutado completamente la aplicación las imágenes intermedias.

La instalación de la aplicación simplemente consiste en copiar el ejecutable del CD, en el disco duro del PC donde se tenga instalado el software que se especifica en los requisitos. Entonces el PC actuará de procesador del sistema de visión de AUTOMAN-TUNEL, si además de los requisitos software dispone del hardware necesario, cámara y tarjeta de adquisición.

Una vez el ejecutable del CD se encuentra en un directorio del disco duro del PC, la aplicación de visión puede ejecutarse con un simple *clic* sobre el ejecutable desde el explorador de Windows situado en el directorio de instalación (también se podría crear un acceso directo desde el *Escritorio*).

Un requisito para la correcta ejecución de la aplicación, es que las imágenes (calibX.bmp) generadas durante el proceso de calibración de la cámara (apartado 7.3.3), se encuentren en el mismo directorio que el ejecutable, si esto no fuera así la aplicación se lanzaría pero fallaría cuando se intentará calibrar la cámara y no permitiría comenzar un proceso de escaneado.

En el directorio en donde se encuentra el ejecutable se almacenan imágenes intermedias que se obtienen durante la fase de extracción de resultados y generación de vistas tras un proceso de escaneado. Estas se corresponden, para cada corte, con la imagen de la vista de distribución de flujo tras aplicar el filtrado con Cany-Derliche (dericheX.bmp), y la imagen tras ecualizar el histograma (hist_equalX.bmp) (ver apartado 5.6.4). Estas imágenes intermedias se irán

machacando cada vez que se ejecute la fase de cálculo de resultados de la aplicación tras un proceso de escaneado, su existencia se explica para permitir análisis posteriores de la vista de flujo desde otras aplicaciones, o entornos como Matlab®, como por ejemplo analizar su histograma. Además, se pueden almacenar las imágenes de las dos vistas (flujo y derivas) de cada corte en el directorio que se desee, a través del interfaz gráfico de la aplicación.

7.2. Instalación del Software de Control

El entorno en el que se ha desarrollado la aplicación de control y en el que se va a ejecutar es distinto. Se ha programado en un PC, en este Proyecto se ha aprovechado el que se ha utilizado para el desarrollo y ejecución de la aplicación de visión, y se ejecuta en el autómatas que controla el sistema de escaneado.

A continuación se describen los requisitos y los pasos necesarios para una correcta instalación de esta aplicación.

7.2.1. Requisitos Previos

Para la instalación y correcta ejecución de la aplicación de control del sistema de escaneado en el autómatas CPU 314C-2DP se debe de:

- Disponer de un PC conectado al autómatas a través de un interfaz MPI. En este Proyecto se ha aprovechado el PC usado también como procesador en el sistema de visión. Este PC tiene instalado la tarjeta CP 6511 que además del interfaz PROFIBUS, dispone del interfaz MPI.
- Tener en el PC el software del CD SIMATIC NET 6.2+SP1 de Simenes®. En concreto el entorno Administrador Simatic (STEP 7).
- La instalación del CD SIMATIC NET 6.2+SP1 exige que el PC tenga instalado WindowsXP con el Service Pack 1.
- Tener configurado el PC como esclavo PROFIBUS-DP, para ver como se puede hacer esto se puede consultar en los tutoriales y manuales que Siemens® provee en [Web 5].

7.2.2. Instalación y Ejecución

El CD del Proyecto contiene una serie de archivos y directorios que conforman un proyecto STEP 7, el cual puede ser abierto desde el Administrador Simatic e instalado en el PC. Este proyecto no solo contiene el código de la aplicación de control, sino que también almacena la configuración como estaciones PROFIBUS-DP del PC y del autómatas.

Para la instalación de la aplicación de visión sobre el autómatas, primero es necesario tener comunicación entre un PC y el autómatas, si se parte de una situación en que el autómatas no está configurado como maestro PROFIBUS-DP es necesario enviarle esta configuración contenida en el proyecto STEP 7 a través de un enlace MPI. Una vez que también se tiene configurado al PC como esclavo PROFIBUS-DP, la comunicación se puede hacer a través de PROFIBUS-DP, entonces sobre la CPU 314C-2 DP se despliega Programa S7(1) y se abren los componentes de Bloques, el bloque OB1 será el que contiene el código de la aplicación de control, haciendo

double clic se abre la herramienta de programación del Administrador Simatic, desde donde se puede enviar el código compilado al autómeta seleccionado *Cargar en Módulo*.

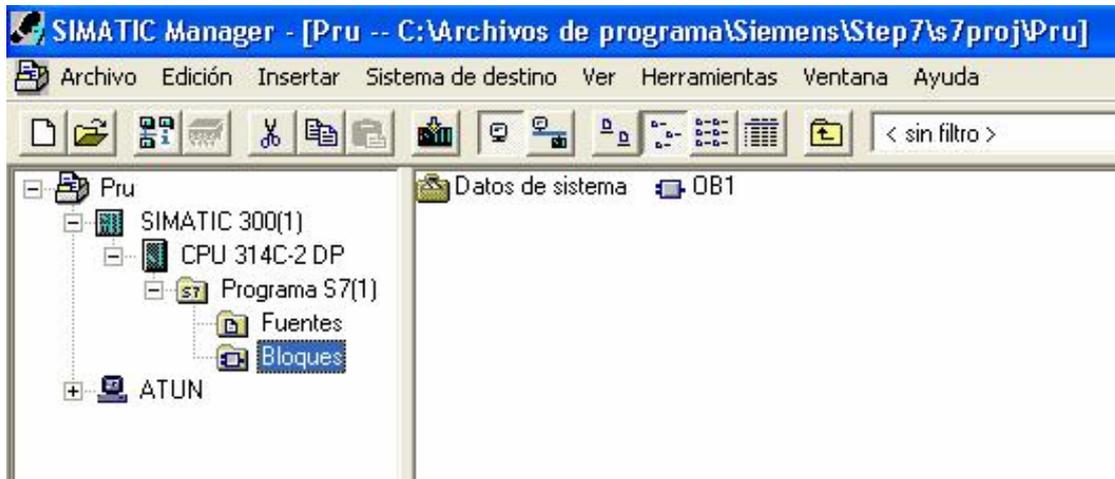


Figura 7. 2. Aspecto del proyecto STEP 7 que contiene la configuración hardware del PC y autómeta, y el código de la aplicación de control.

La ejecución de la aplicación es automática una vez que se haya cargado el bloque sobre el autómeta y el autómeta se encuentre activado. Para que esté activado el interruptor del autómeta ha de estar en posición RUN, la verificación de que efectivamente está activado se hace a través de los LEDs del *display de estado*, en el apartado 4.6.1 se describe como funciona este display.

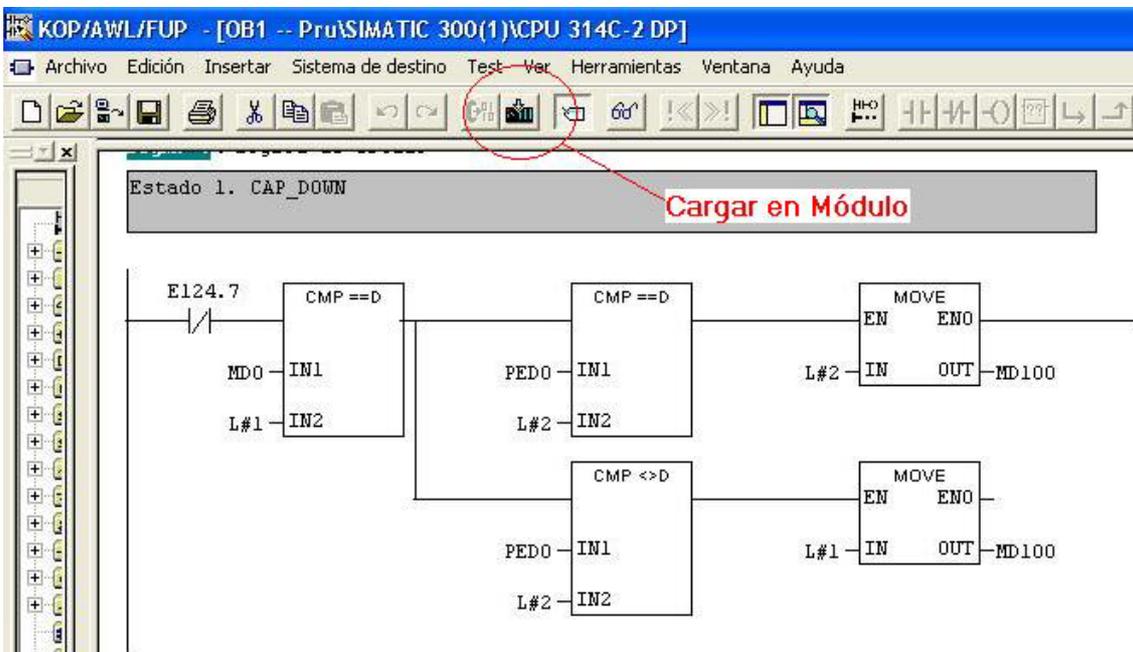


Figura 7. 3. Herramienta de programación Simatic.

7.3. Manual de Uso de la Aplicación de Visión

En este apartado se describe como utilizar correctamente la aplicación que controla al sistema de visión de AUTOMAN-TUNEL. Se parte de una correcta instalación sobre el PC que actúa como procesador del sistema de visión, y de que la aplicación ya se encuentra lanzada. Primero se explica como usar el interfaz gráfico de la aplicación, después se explican los posibles errores que pueden surgir durante la ejecución junto con sus posibles causas y soluciones. A final se describe como se han de generar las imágenes que la aplicación toma para calibrar la cámara del sistema de visión.

7.3.1. Manejo del Interfaz Gráfico

Una vez lanzada la aplicación de visión, *ScanTunnel*, tiene el aspecto de la figura 7.4. El interfaz gráfico de la aplicación está compuesto por una imagen y debajo dos agrupaciones de botones, la de la izquierda contiene los botones que controlan la funcionalidad de la aplicación, y la de la derecha contiene botones que permiten seleccionar el corte que se desea ver y guardar las imágenes resultado en directorio que se desee.

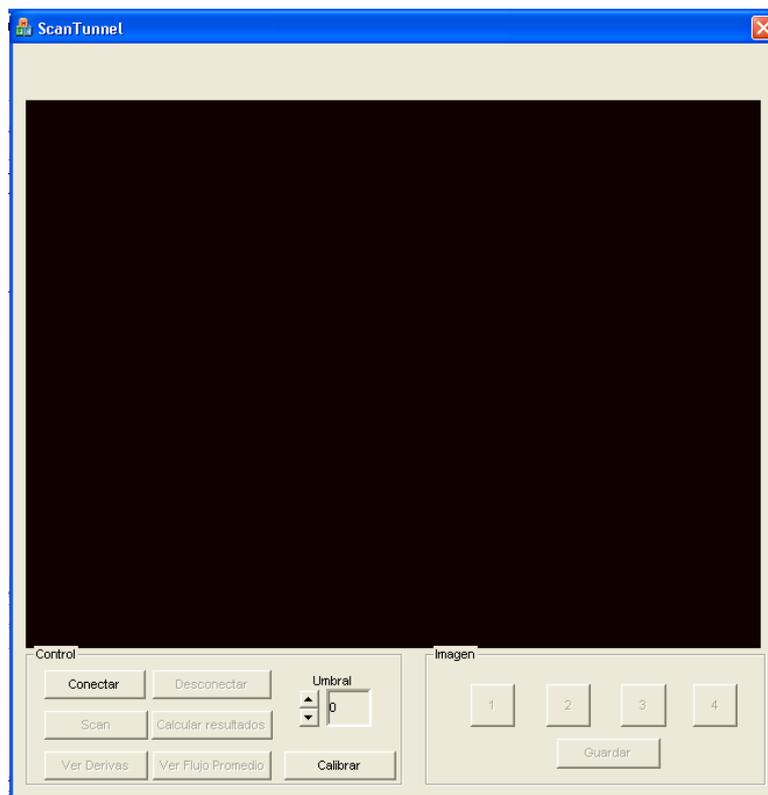


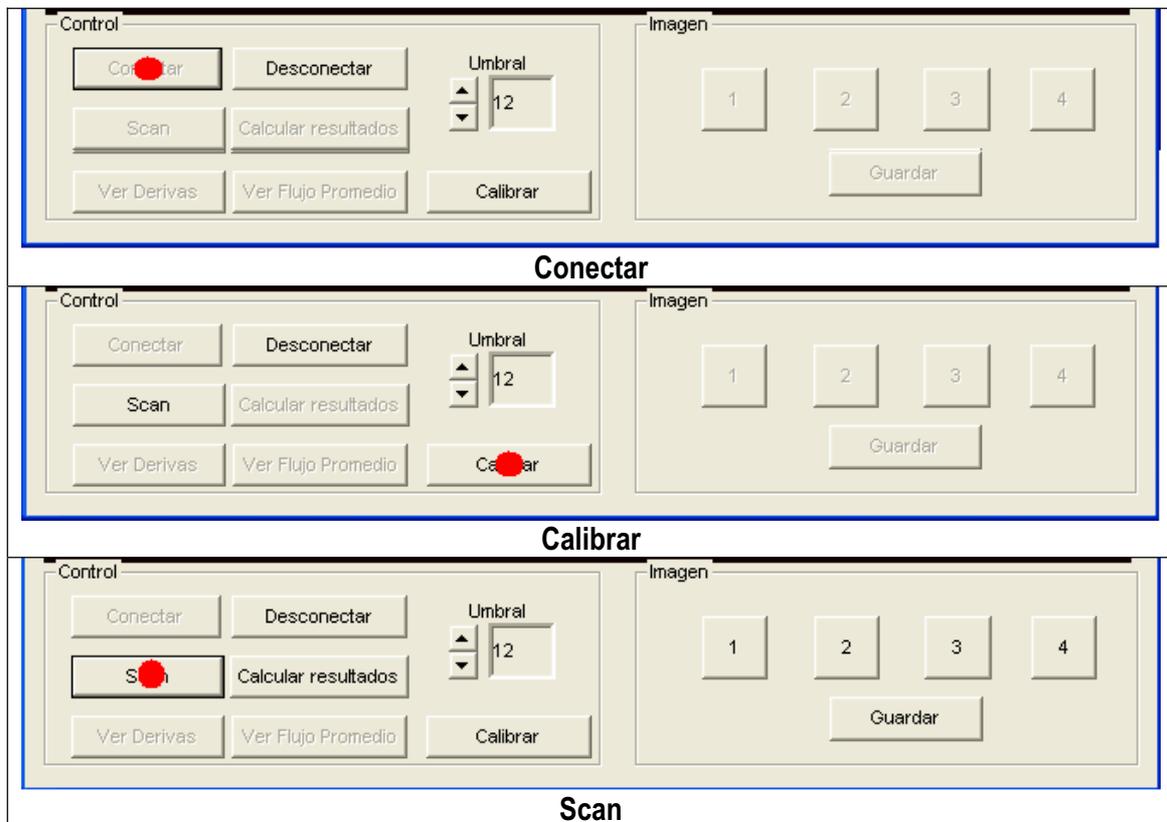
Figura 7. 4. Aspecto de la aplicación de visión al ser arrancada.

Los botones de la aplicación y su funcionalidad se resumen a continuación:

- **Conectar:** conecta a la aplicación con el servidor OPC. Este botón está habilitado sólo si no se está conectado al servidor.
- **Desconectar:** desconecta a la aplicación del servidor OPC, solo está habilitado si se está conectado al servidor.

- **Calibrar:** calibra la cámara tomando como partida las imágenes de calibración. Siempre está habilitado.
- **Umbral:** en realidad no se trata de un botón sino de un *Spin Control*, usado para poder modificar el valor umbralización que se utilizará durante el proceso de escaneado entre sus valores permitidos (0, 1, 2,..., 255). Siempre está habilitado.
- **Scan:** realiza un proceso de escaneado. Solo se podrá realizar un escaneado si se está conectado al servidor OPC.
- **Calcular Resultados:** obtiene los resultados numéricos de las derivas y genera las vistas resultado, vista del flujo de líquido y de las derivas. Este botón solo está habilitado si se ha hecho un escaneado.
- **Ver Derivas:** ofrece la vista de las derivas. Se habilita tras la generación de las vistas.
- **Ver Flujo Promedio:** ofrece la vista de flujo promedio. Se habilita tras la generación de las vistas.
- **1, 2, 3, 4:** cambian el corte que se está mostrando.

Durante toda su ejecución, para facilitar el manejo por parte del usuario y evitar errores, solo se encontrarán habilitados los botones que en ese momento se pueden usar, así en la tabla 7.1 se muestra una posible secuencia pasos que partiendo del estado inicial (figura 7.4) lleva a la obtención de los resultados que se mostrarán en forma de vistas (imágenes).



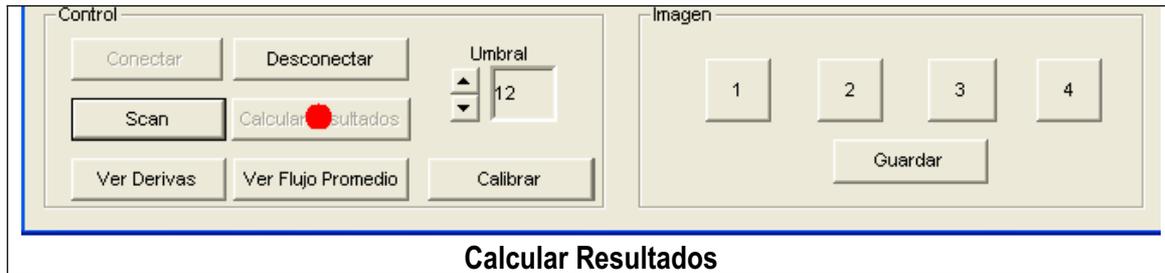


Tabla 7. 1. Ejemplo de pasos de ejecución de la aplicación para la obtención de resultados, la situación del interfaz mostrado se corresponde a cuando se ha terminado de ejecutar la funcionalidad del botón pulsado.

Para poder tomar imágenes durante el proceso de escaneado es necesario que previamente se haya calibrado la cámara, para calibrar es necesario que las imágenes de calibración se encuentren en el directorio donde se encuentra el ejecutable de la aplicación. Una vez terminado el proceso de escaneado se pueden calcular los resultados, que una vez obtenidos se generan las dos vistas en las que estos se mostrarán. Por defecto, se muestra la vista de derivas del corte 1 (inferior), el cambio de vista o de corte se realiza pulsando los botones dedicados a ello. Dadas las condiciones de iluminación especificadas en este Proyecto el valor umbral que se recomienda utilizar es 12, si estas condiciones cambian, lo que podría pasar si por ejemplo se modifica la impermeabilización frente a la luz exterior del cajón, el valor que permita una correcta obtención de resultados puede que deba ser modificado.

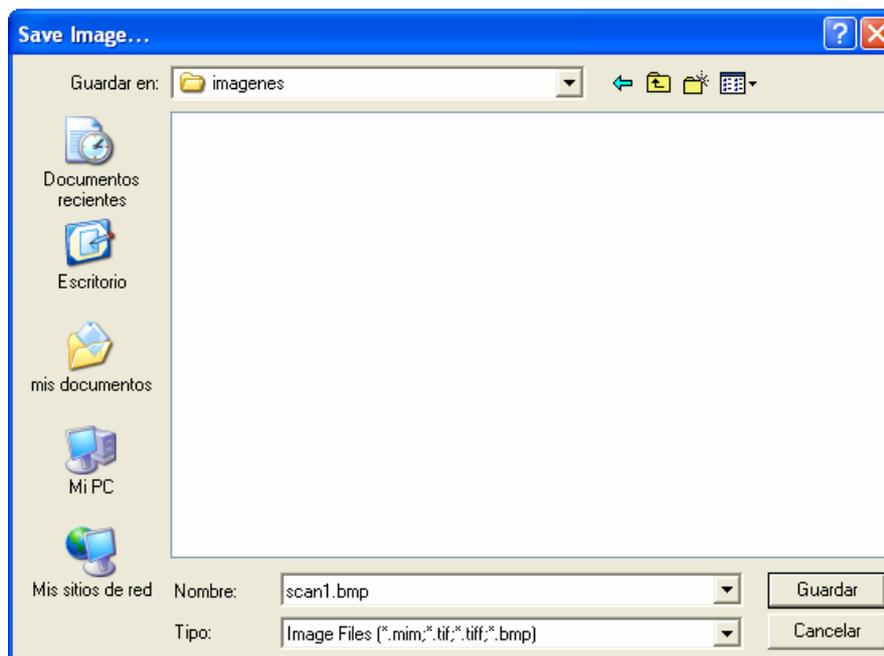


Figura 7. 5. Diálogo para especificar el directorio donde se almacenarán las imágenes resultado.

Tras la etapa de cálculo de resultados se generan las imágenes intermedias que serán almacenadas en el directorio de la aplicación, para almacenar los resultados definitivos compuestos por dos imágenes (una para cada vista) por cada corte (en total 8 imágenes), se utiliza el botón *Guardar*. Este botón abre un diálogo en el que se ha de especificar el nombre de la imagen de la vista de flujo del corte 1, para seguir con la notación que tomarán el resto de imágenes se recomienda almacenarla con al nombre *scan1.bmp*, el resto de imágenes se almacenan automáticamente en ese mismo directorio. Todas las imágenes que guarda esta

aplicación lo harán en formato BMP, que al no estar comprimido no distorsiona las imágenes y al ser estándar de Windows éstas pueden ser abiertas por otras aplicaciones, como el Visor de Windows, Matlab, etc.

Para terminar de ejecutar la aplicación basta con hacer *clic* sobre el aspa situada en la esquina superior derecha del interfaz de la aplicación.

7.3.2. Errores

La aplicación puesto que ha de comunicarse con elementos externos, sistema de escaneado vía servidor OPC, y tomar imágenes del disco duro para realizar la calibración es susceptible de sufrir errores, bien por un fallo en el sistema de escaneado o en la comunicación con éste ó bien porque no estén las imágenes de calibración o éstas estén mal tomadas. La aplicación es capaz, hasta cierto punto, de identificar los errores de manera que informa al usuario de las posibles causas del error, para que de esta forma el usuario sea capaz de corregirlas y, si es posible, se pueda continuar con la ejecución de la aplicación.

Los errores son mostrados mediante la presentación de un diálogo de error que aparecerá sobre el diálogo principal de la aplicación. Pulsando *Aceptar* el usuario notifica a la aplicación que conoce el fallo, y ha adoptado las medidas oportunas para que el fallo no se vuelva a repetir, entonces se reanuda la actividad de la aplicación.

Errores de conexión

Este tipo de error ocurre cuando existen problemas en la comunicación de la aplicación de visión con la aplicación de control.


<ul style="list-style-type: none">• No se puede acceder al servidor OPC.• Asegurarse de que el servidor OPC SimaticNET está arrancado en el PC, y si no arrancarlo a través del software Simatic.

<ul style="list-style-type: none">• Error en el autómatas.• Posiblemente se deba a que el autómatas este apagado, en estado STOP o incluso desconectado del bus. Se soluciona encendiendo, arrancando o conectándolo al bus.


<ul style="list-style-type: none"> • El autómata no se reseteo al comenzar el proceso de escaneado • Suele deberse a que el sistema de escaneado no se encontraba en su posición de inicio al comenzar el escaneado, esto podría deberse a que un escaneado previo no finalizó correctamente. Se resetea manualmente al autómata, y si además el láser no se encuentra en la posición inicial (corte inferior), mediante el modo manual de la aplicación de control (apartado 7.4.1) se coloca en esta posición.

Tabla 7. 2. Causas y soluciones a los errores de conexión.

Errores externos

Suelen ocurrir por fallos en el sistema de escaneado.


<ul style="list-style-type: none"> • Fallo en los sensores o el actuador el sistema de escaneado. • Revisar la conexión, posición y estado de los sensores, sobre todo del sensor de ultrasonidos, revisar las conexiones de la caja de electroválvulas, y asegurarse de que a este la llega presión del motor.

<ul style="list-style-type: none"> • Elección incorrecta del umbral o fallo del sistema de escaneado. • Asegurarse de que la fuente de iluminación (láser) está encendida y su posición sobre el vástago del pistón neumático es correcta, sino asegurarse de que el sistema de escaneado realiza las paradas en las posiciones correctas. Otra causa podría ser la elección de un valor de umbral inadecuado, si la impermeabilización frente a la luz externa del cajón es correcta se recomienda usar el valor 12. Una oclusión o defecto en la óptica de la cámara también podría generarlo.

Tabla 7. 3. Causas y soluciones a los errores externos.

Errores ActiveMIL

Son generados directamente por el código de ActiveMIL, la aplicación de visión solo contempla la generación de estos errores durante la etapa de calibrado, la generación de algún error de este tipo que no está contemplada por la aplicación de visión podría deberse a fallos en el hardware del sistema de visión.

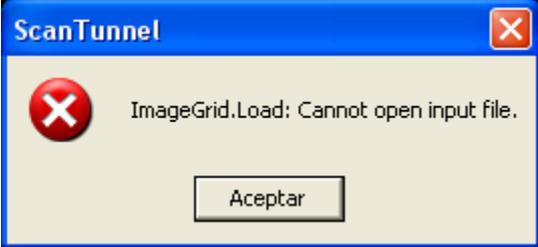
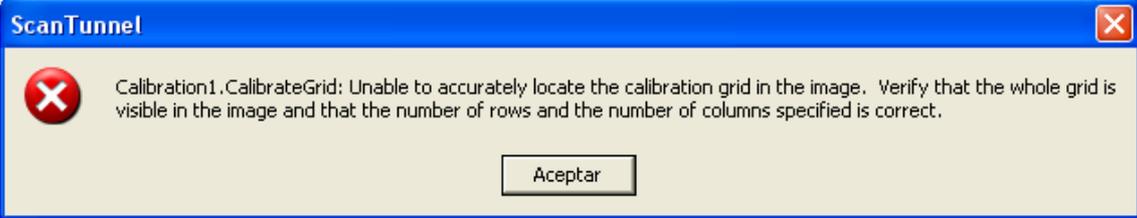
 <ul style="list-style-type: none"> • Falta alguna de las imágenes de calibración. • Asegurarse que todas las imágenes de calibración se encuentran en el directorio de la aplicación, y tienen los nombres correctos.
 <ul style="list-style-type: none"> • Alguna de las imágenes de calibración no se ha generado correctamente. • Generar las imágenes de calibración de acuerdo a como se explica en el apartado 7.3.3.

Tabla 7. 4. Errores generados por ActiveMIL contemplados por la aplicación de visión.

7.3.3. Generación de Imágenes para la Calibración

A continuación se expone una serie de pautas que tratan de asegurar la generación de imágenes para la calibración, de manera que no se genere ningún error durante la etapa de calibración ó durante la ejecución de la aplicación de visión.

La rejilla de calibración ha de estar formada por una superficie plana de círculos que deberían de poseer las siguientes características [8]:

- El radio de los círculos de la rejilla debería estar entre 6 y 10 píxeles.
- La distancia entre los círculos de la rejilla debería moverse entre los 18 y 32 píxeles.
- La distancia mínima entre los bordes de los círculos debería ser 6 píxeles o más.
- La imagen del píxel debería tener un alto contraste.

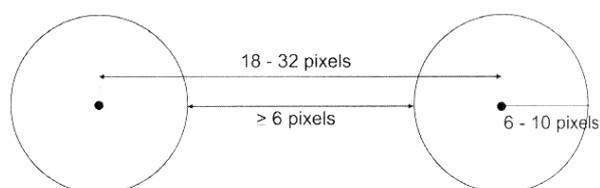


Figura 7. 6. Restricciones que ha de cumplir los círculos de la imagen de la rejilla de calibración.

Se ha de generar una imagen para cada corte y se nombrarán como *calibX.bmp*, donde *X* denota el número de corte. La posición de la rejilla en cada corte ha de ser tal que esté contenida en el plano que define el sistema de iluminación para ese corte y sea visible desde la cámara, esto se puede conseguir colocando la rejilla alineada con el haz de luz de láser plano. Cuanto mejor sea la alineación entre haz láser y rejilla más precisa será la calibración, y después se obtendrán resultados más ajustados a la realidad. Las imágenes pueden ser capturadas y almacenadas en disco con Matrox® Intelicam [16], aplicación disponible cuando se instala el CD MIL 8.0 que contiene ActiveMIL.

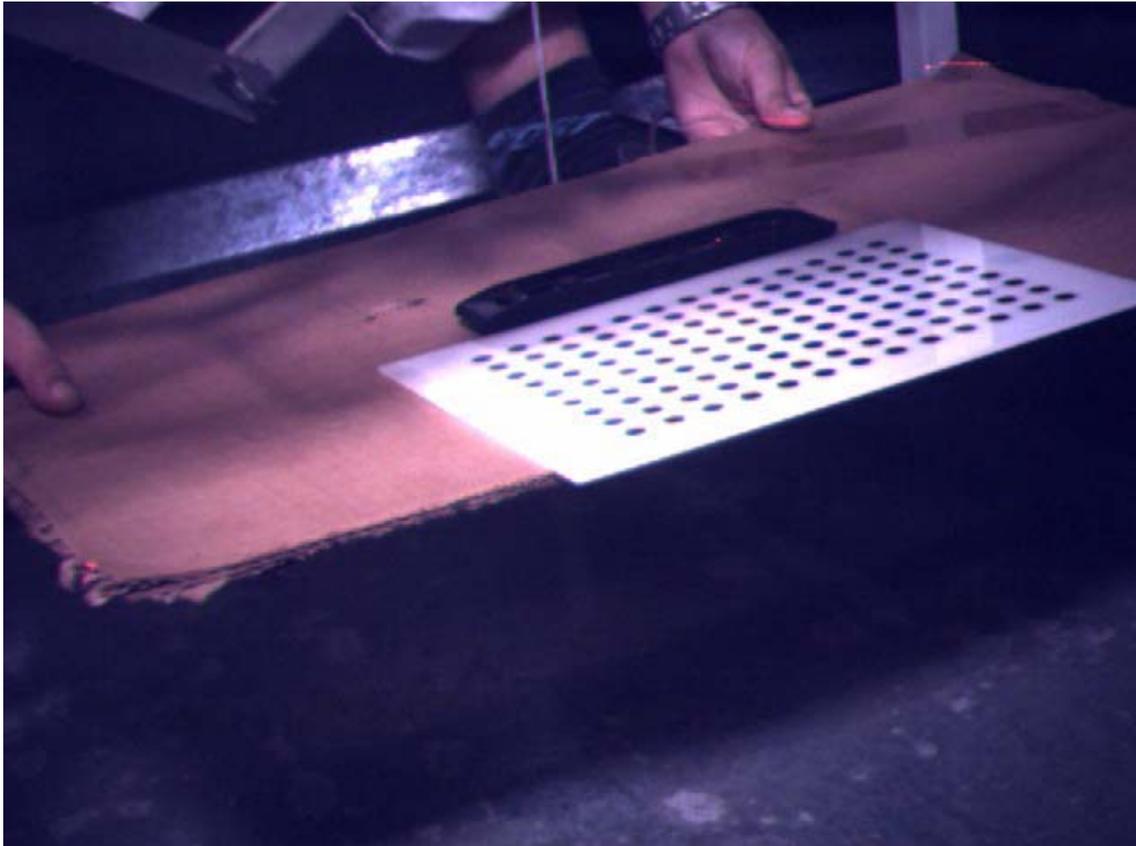


Figura 7. 7. Colocación de la rejilla de calibración alineada con el haz láser.

No es necesario que la rejilla ocupe completamente la región trabajo, ya que la aplicación de visión realiza una calibración en la que solo se compensa la distorsión por perspectiva y no se tiene en cuenta las distorsiones de la óptica de la cámara. En este tipo de calibración se obtiene precisión, obviando la distorsión de la óptica, dentro y fuera de la región de trabajo, y comenzaría a funcionar con solo 4 puntos, aunque a mayor cantidad de puntos mayor precisión.

Para cada corte la cantidad de puntos y la distancia entre puntos es distinta. Esto se debe a que para asegurar la convergencia de los algoritmos de calibración en imágenes en las que la cámara está muy desalineada con el plano objeto, como es el caso de los cortes superiores, la cantidad de puntos ha de ser pequeña, sin embargo, en el corte inferior se puede usar una cantidad de puntos mayor, para mejorar la precisión, y el algoritmo sigue convergiendo. Para cada corte las especificaciones de cada rejilla son:

- Corte 1 (inferior): 9 filas separadas (distancia entre los centros de los círculos) 1.9cm y 13 columnas separadas 1.9cm.

- Corte 2: 3 filas separadas 7.6cm y 5 columnas separadas 5.6cm.
- Corte 3: 3 filas separadas 7.6cm y 5 columnas separadas 5.6cm.
- Corte 4 (superior): 2 filas separadas 14.7cm y 3 columnas separadas 11.1cm.

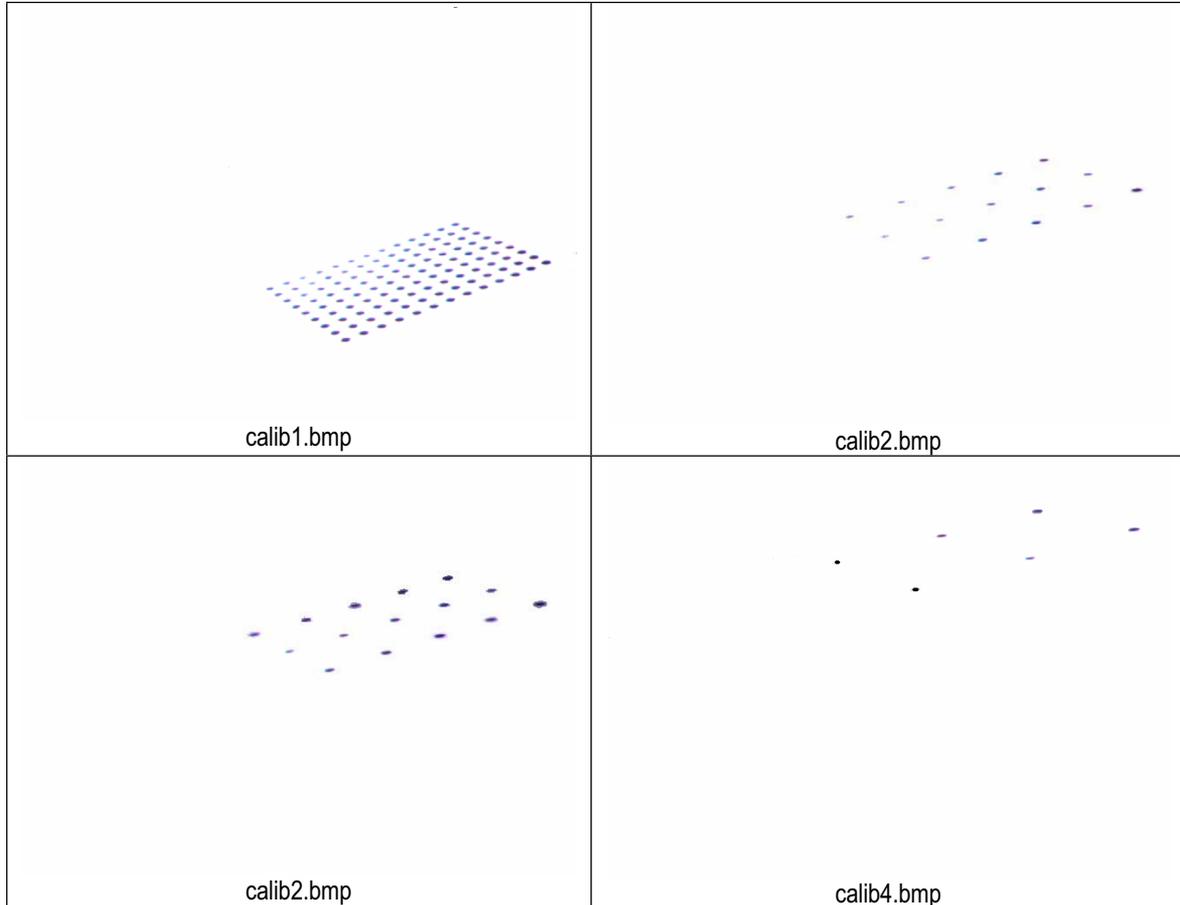


Figura 7. 8. Ejemplos de imágenes de calibración correctas de los 4 cortes.

No es recomendable emplear las imágenes directamente tal y como se capturan, en una alta proporción suelen generar errores durante la calibración, sobre todo en las imágenes de los cortes superiores. Esto se debe a que debido a la forma del cajón y a la colocación de la cámara la rejilla no puede ocupar toda la imagen capturada (ver figura 7.7), y pudieran aparecer objetos en la imagen que el algoritmos de calibración interpretara como círculos de la rejilla. Otra causa, es que a veces, en condiciones de iluminación deficientes, pudiera ser necesario aumentar el contraste entre los círculos y la rejilla. Los retoques sobre las imágenes se pueden realizar de forma manual mediante cualquier editor de imágenes (ver figura 7.8).

7.4. Manual de Uso de la Aplicación de Control

En principio no tendría sentido hablar de manual de uso de la aplicación de control, ya en su *modo de ejecución normal* es la aplicación de visión la que se encarga de controlar a esta aplicación a través de su comunicación con ésta. No obstante, esta aplicación posee un *modo de*

ejecución manual en el que el usuario puede controlar directamente la altitud del sistema de escaneado, lo cual se puede aprovechar para ajustar su rango.

7.4.1. Manejo Manual del Sistema de Escaneado

A este modo de ejecución se accede activando el interruptor E124.7 del tablero eléctrico del autómat, el cual está conectado con la entrada digital 124.7. En este modo la aplicación de control deja de escuchar y enviarle información a la aplicación de visión, lo que significa que si estuviera en medio de un proceso de escaneado, éste se detendría.

En este modo solo se puede subir y bajar la fuente de iluminación del sistema de escaneado. Para bajar basta con activar el interruptor E125.0 y para subir E125.1, conectados respectivamente con las entradas digitales 125.0 y 125.1. Se identifica cuando se ha alcanzado los extremos del rango definido por los sensores de proximidad, cuando se activan los LEDs del autómat correspondientes a las salidas digitales 124.0 (inferior), 124.1 (superior).

7.4.2. Ajuste del Rango de Escaneado

Es posible que se desee cambiar el rango de escaneado, por ejemplo si se ha cambiado la distancia sobre el suelo del cajón del pulverizador. Es importante verificar que el nuevo rango es factible, es decir, este dentro del rango del cilindro neumático y dentro de él ni el sensor de ultrasonidos ni el láser sufren ninguna colisión.

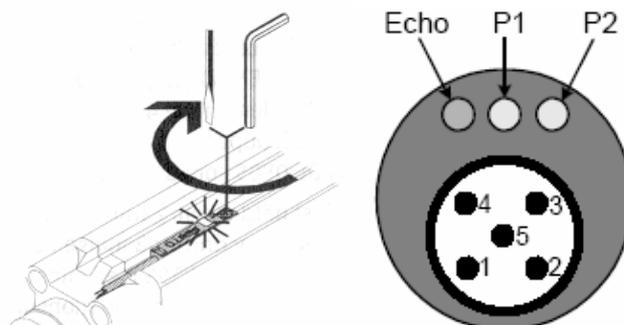


Figura 5. 30. Fijación de un sensor de proximidad al cilindro neumático (izquierda), y LEDs del sensor de ultrasonidos (derecha).

El rango sobre el que se moverá el sistema de escaneado viene definido por la posición de los sensores de proximidad sobre el cilindro neumático, por lo que habrá que cambiar su posición. También, es necesario ajustar el rango del sensor de ultrasonidos con el nuevo rango definido por los sensores de proximidad, así las dos paradas intermedias del sistema de escaneado estarán situadas a la misma distancia entre sí que con la parada extrema (inferior o superior) más próxima, con lo que se consigue repartir uniformemente las paradas a lo largo del rango. El movimiento del sensor de ultrasonidos, colocado en el extremo del vástago del cilindro, a las posiciones necesarias para programarlo, se realiza mediante el modo manual de la aplicación de control del autómat. Entonces, el proceso de ajuste de rango de escaneado contiene los siguientes pasos:

1) Ajustar la posición de los sensores de proximidad: estos sensores están fijados a una ranura del cilindro diseñada especialmente para alojar a este tipo de sensores, para moverlos

basta con girar a la izquierda el tornillo de fijación, mover el sensor a la posición deseada y volver fijar el sensor girando el tornillo de fijación hacia la derecha.

2) Definir extremo inferior del sensor de ultrasonidos: se conecta el conector correspondiente al pin *Teach-In* (hilo rosa), con la tierra (GND) de la fuente de alimentación del tablero eléctrico del autómatas, durante 8 segundos hasta que los LEDs P1 y *Echo* parpaddeen simultáneamente (parpadearán con una frecuencia de 2Hz). Entonces, se quita el contacto entrando el sensor en el modo Teach para P1 (4mA, se toma como extremo inferior):

- El LED P1 cambia su frecuencia de parpadeo a 1Hz y el LED Echo retoma a su función habitual de mostrar si el objetivo está correctamente alineado.
- Se tiene un tiempo de un minuto para grabar la posición P1.
- Se coloca el sensor de ultrasonidos en el extremo inferior del nuevo rango. Cuando se alcanza esta posición se enciende el LED del autómatas correspondiente a la salida digital 124.0.
- Conectar y Desconectar Teach-In con GND, almacenándose el nuevo valor para P1 y retornando el sensor a su modo de funcionamiento normal.

3) Definir el extremo superior del sensor de ultrasonidos: Conectar Teach-In con GND durante 13 segundos, hasta que los LEDs P2 y Echo comiencen a parpadear simultáneamente a 2Hz. Nótese que antes de alcanzar el modo Teach para P2 se pasará por el modo Teach para P1, a los 8 segundos P1 parpadeará a 2Hz, entonces se ha de esperar 5 segundos adicionales para que P2 comience a parpadear. Ahora, se desconecta Teach-In entrando el sensor en el modo Teach para P2:

- El LED P2 cambia su frecuencia de parpadeo a 1Hz y el LED Echo retorna a su función habitual de mostrar si el objetivo está correctamente alineado.
- Se tiene un tiempo de un minuto para grabar la posición P2.
- Se coloca el sensor de ultrasonidos en el extremo superior del nuevo rango. Cuando se alcanza esta posición se enciende el LED del autómatas correspondiente a la salida digital 124.1.
- Conectar y Desconectar Teach-In con GND, almacenándose el nuevo valor para P2 y retornando el sensor a su modo de funcionamiento normal.

CAPÍTULO 8

RESULTADOS, CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se describen las aportaciones del sistema AUTOMAN-TUNEL desarrollado en este Proyecto, las cuales han tratado de cubrir los objetivos marcados inicialmente, además de otras aportaciones que se han ido incorporando y que en un principio no estaban previstas.

Durante el desarrollo del sistema se ha ido adquiriendo un conocimiento más profundo del problema del que se tenía en un principio, de modo que se han podido concluir formas de mejorar los diseños iniciales del Proyecto, de las que parte se han incorporado en forma de aportaciones adicionales, y otra parte se proponen como trabajos futuros que mejorarán el comportamiento del sistema.

Este capítulo se acompaña también de un estudio sobre un pulverizador. El objetivo de este estudio, más que tratar de definir las características y prestaciones del pulverizador, lo que intenta es demostrar el correcto funcionamiento del sistema implementado en este Proyecto y su utilidad.

8.1. Estudio sobre un pulverizador

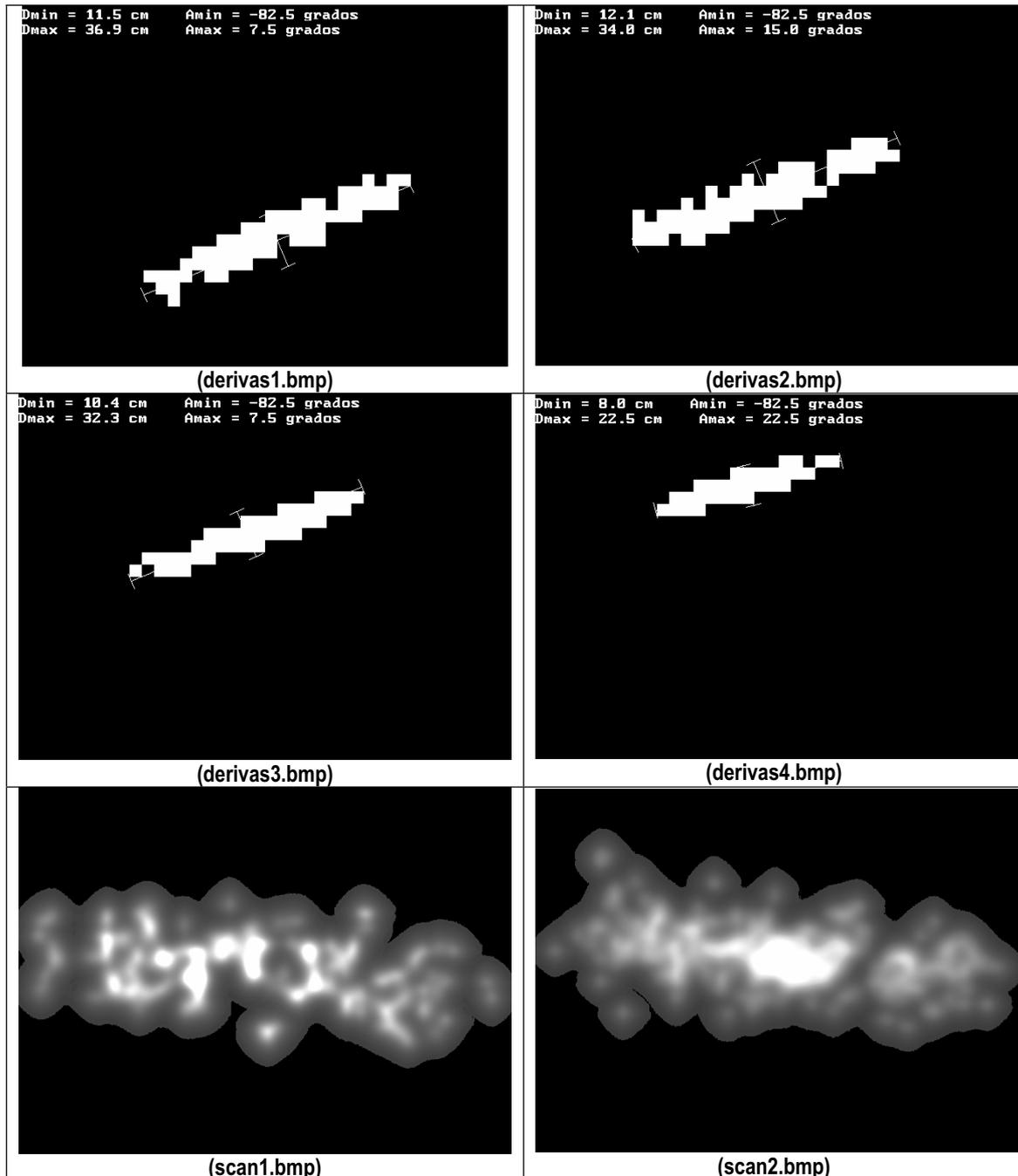
Se ha tomado un pulverizador de prueba que ha sido conectado al sistema de inyección de AUTOMAN-TUNEL, este estudio tratará de averiguar como es la nube de pulverización que este pulverizador genera, definiendo los límites máximos de la nube, así como la distribución de líquido en su interior.

El estudio también tratará de averiguar que factores afectan a la forma y distribución de la nube de pulverización. Así como establecer de forma cualitativa, con el objetivo de corroborar el

correcto funcionamiento del sistema y no el de extraer su relación exacta, como es esta dependencia.

8.1.1. Estudio de una Medida

En este apartado se realizará el estudio de los resultados de obtenidos durante un escaneado en ausencia de viento, con una presión de 1bar, y con un tiempo de promediado de 3.6 segundos (110 frames). Con estos parámetros y para el pulverizador fijado, la aplicación de visión genera los siguientes resultados:



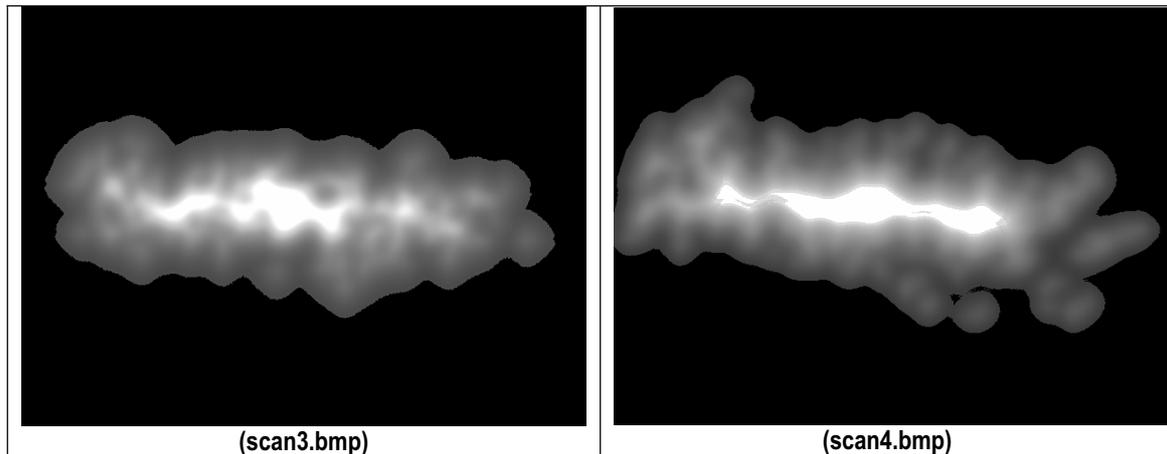


Tabla 8. 1. Resultado del escanear sin viento, 1bar de presión y 3.6 segundos de tiempo de promediado.

De la tabla 8.1 se pueden extraer varias conclusiones. La primera, se puede extraer de los resultados sobre la deriva (imágenes derivaX.bmp), observándose como ésta, tanto la máxima como la mínima, aumenta en cada corte conforme éstos se alejan del pulverizador. Este resultado es, sin duda, lo esperado, ya que el propio objetivo del pulverizador es expulsar el líquido en forma de gotas con ciertos ángulos de dispersión (dependiendo del eje que se tome), de forma que las gotas pulverizadas se vayan abriendo conforme se alejan del pulverizador para regar una zona más o menos amplia en función de la distancia al pulverizador.

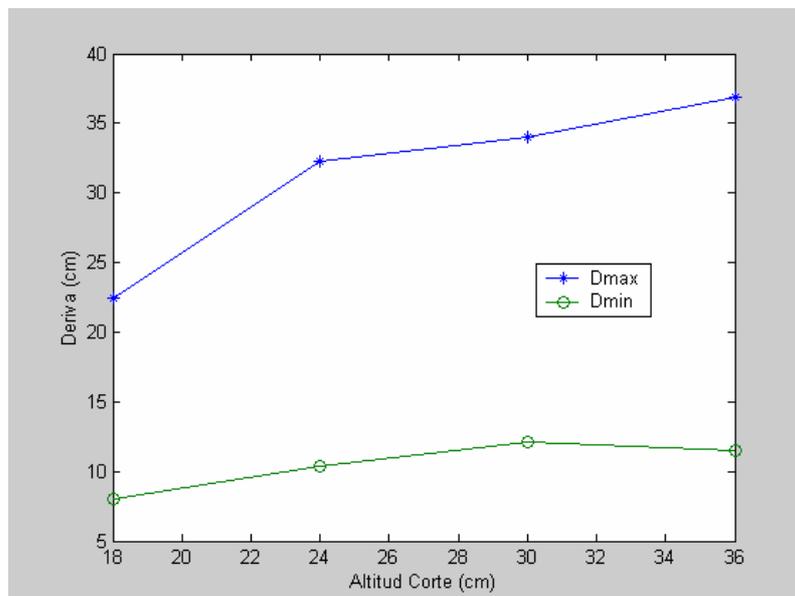


Figura 8. 1. Derivas para cada corte.

Las gotas expulsadas por el pulverizador, según el modelo de la cinemática clásica, seguirán una trayectoria parabólica debido a que su aceleración estará gobernada por dos componentes; el propio peso de la gota que la arrastrará en dirección perpendicular al suelo con una aceleración debida a la gravedad, y la fuerza con que el pulverizador expulsa la gota a una determinada velocidad. Según la dirección y el módulo de la fuerza de expulsión del pulverizador, que depende del propio pulverizador y de la presión con que se alimente, el

moviendo compuesto se alejará más o menos del movimiento que rectilíneo define el peso de la gota.

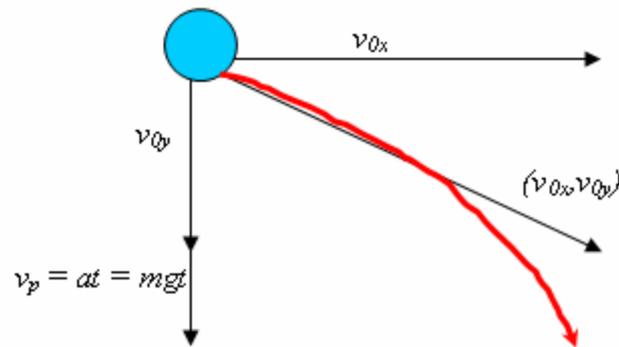


Figura 8. 2. Diagrama de componentes de velocidad que lleva una gota y que definen un movimiento parabólico.

Entonces la altura (h) de una gota y la deriva (distancia de la gota a la perpendicular entre el suelo y pulverizador, d) en cada instante vendrán gobernadas por la siguiente expresión:

$$h(t) = h_0 - v_{0y}t - \frac{1}{2}mgt^2 \quad (8.1)$$

$$d(t) = v_{0x}t \quad (8.2)$$

Según este modelo de movimiento para las gotas y considerando que la deriva es definida por las gotas más externas de la región de pulverización, el valor de la deriva en función de la altura se podría estimar a partir de los resultados obtenidos mediante una aproximación cuadrática que emulase la trayectoria seguida por estas gotas.

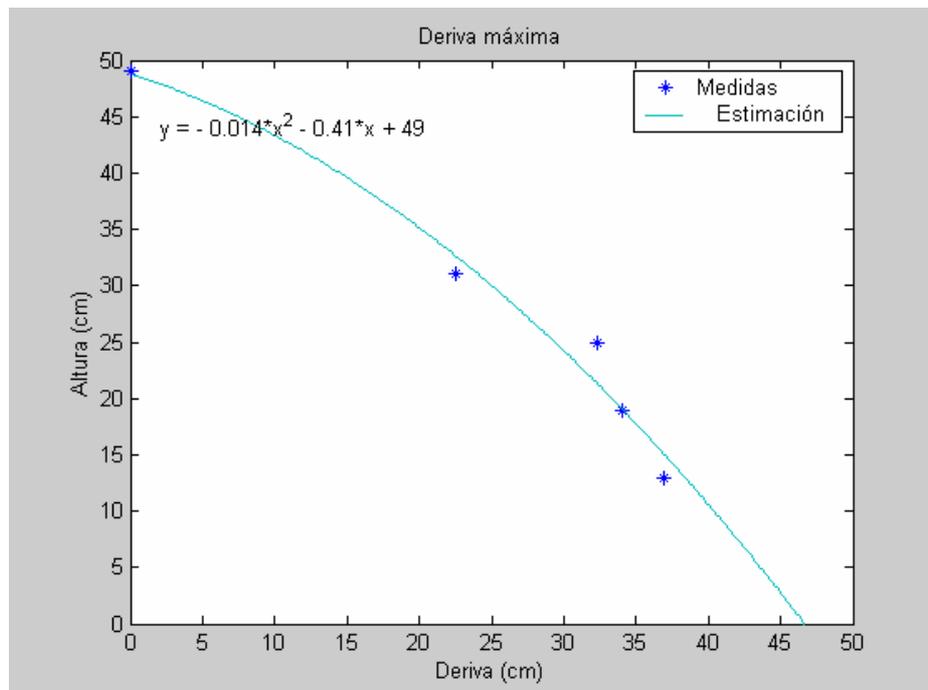


Figura 8. 3. Estimación de la deriva máxima en función de la altura.

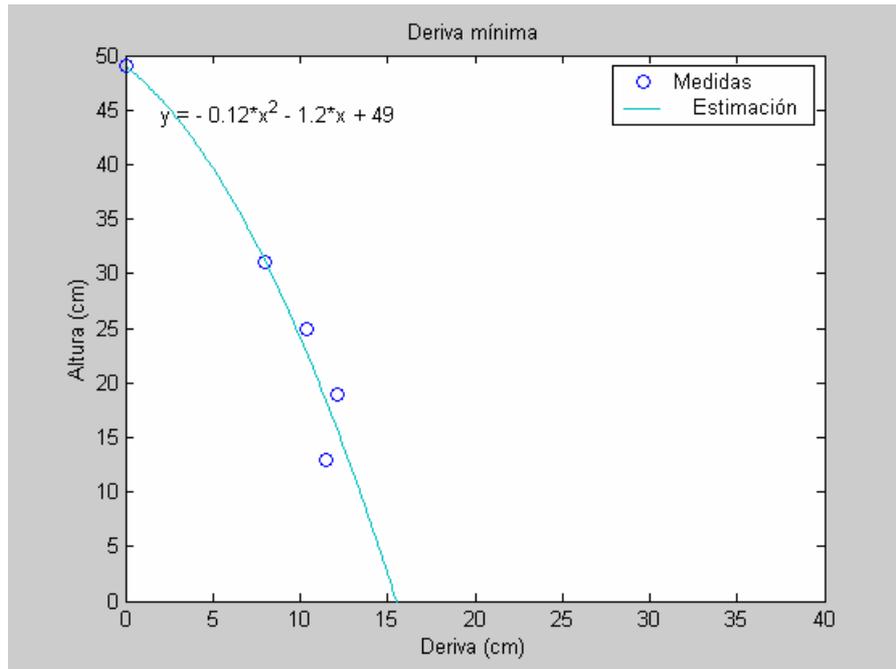


Figura 8. 4. Estimación de la deriva mínima en función de la altura, en el suelo esta deriva sería de poco más de 15 cm.

Con respecto a la distribución de flujo también se pueden extraer conclusiones, la principal es que éste se va dispersando sobre áreas mayores perdiendo su homogeneidad. Éste es un aspecto importante ya que, por ejemplo para el prototipo AUTOMAN, se buscarán pulverizadores que sobre las plantas rocíen el herbicida lo más uniformemente posible, a fin de evitar que éste se malgaste en unas zonas y falte en otras.

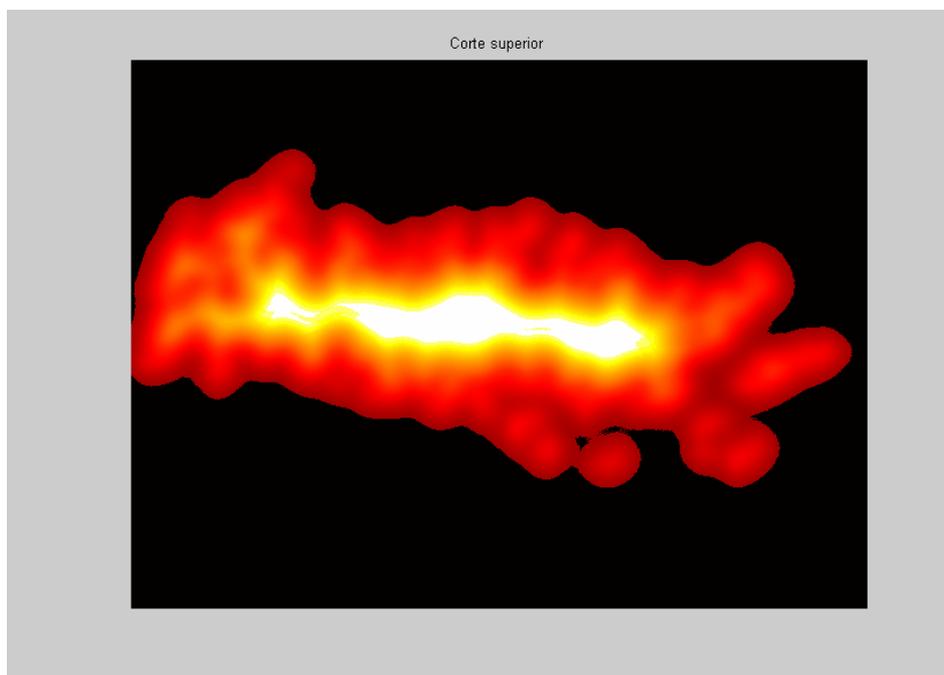


Figura 8. 5. Distribución de flujo del corte superior.

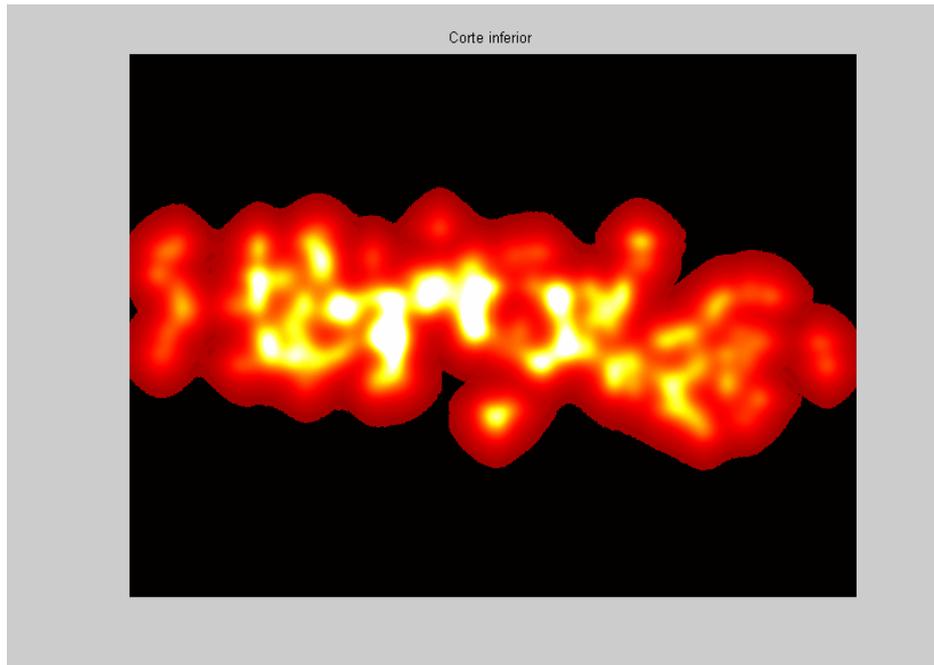


Figura 8. 6. Distribución de flujo del corte inferior.

8.1.2. Resultados en función de la Presión de Inyección

Cuanta más presión sobre el pulverizador, mayor velocidad inicial conferirá a las gotas en la dirección que éste marca, por lo que esta componente tomará un mayor peso en la trayectoria parabólica de la gota ocultando el efecto de la gravedad.

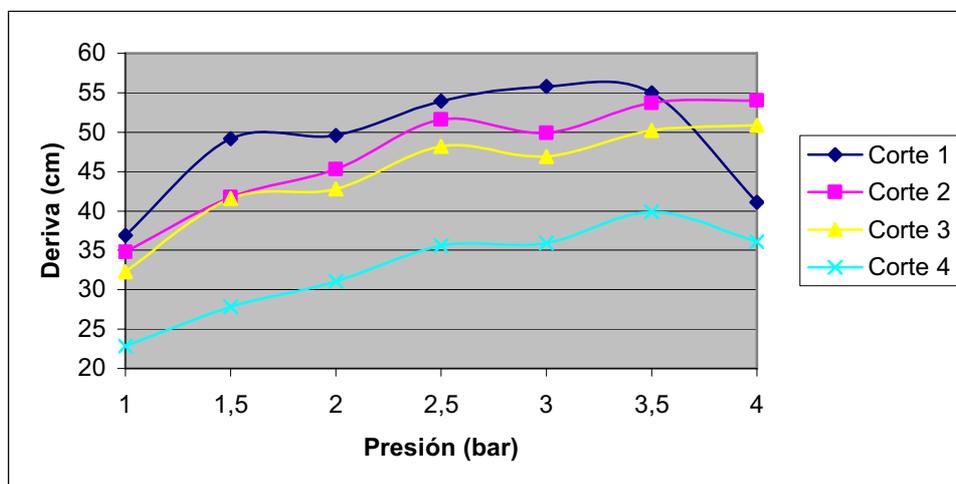


Figura 8. 7. Variación de la deriva máxima en función de la presión para cada corte.

En la figura 8.7 haber una tendencia hacia el incremento de la deriva en función del incremento de presión, que parece truncarse cuando se alcanzan los 4 bares de presión. La medida de deriva sobre 4bar para el corte 1 parece un caso claro de error de medida (*outlier*), posiblemente originado por un error en la aplicación de visión. Aún así, tomando el resto de cortes para la presión de 4bar se confirma una estabilización de las derivas.

En cuanto a la variación en la distribución de flujo con la presión y siempre en referencia al corte inferior, parece observarse que mientras a presiones bajas el flujo se distribuye con cierta granularidad homogénea, al aumentar la presión el flujo tiende a acumularse en una región concreta (ver figura 8.8).

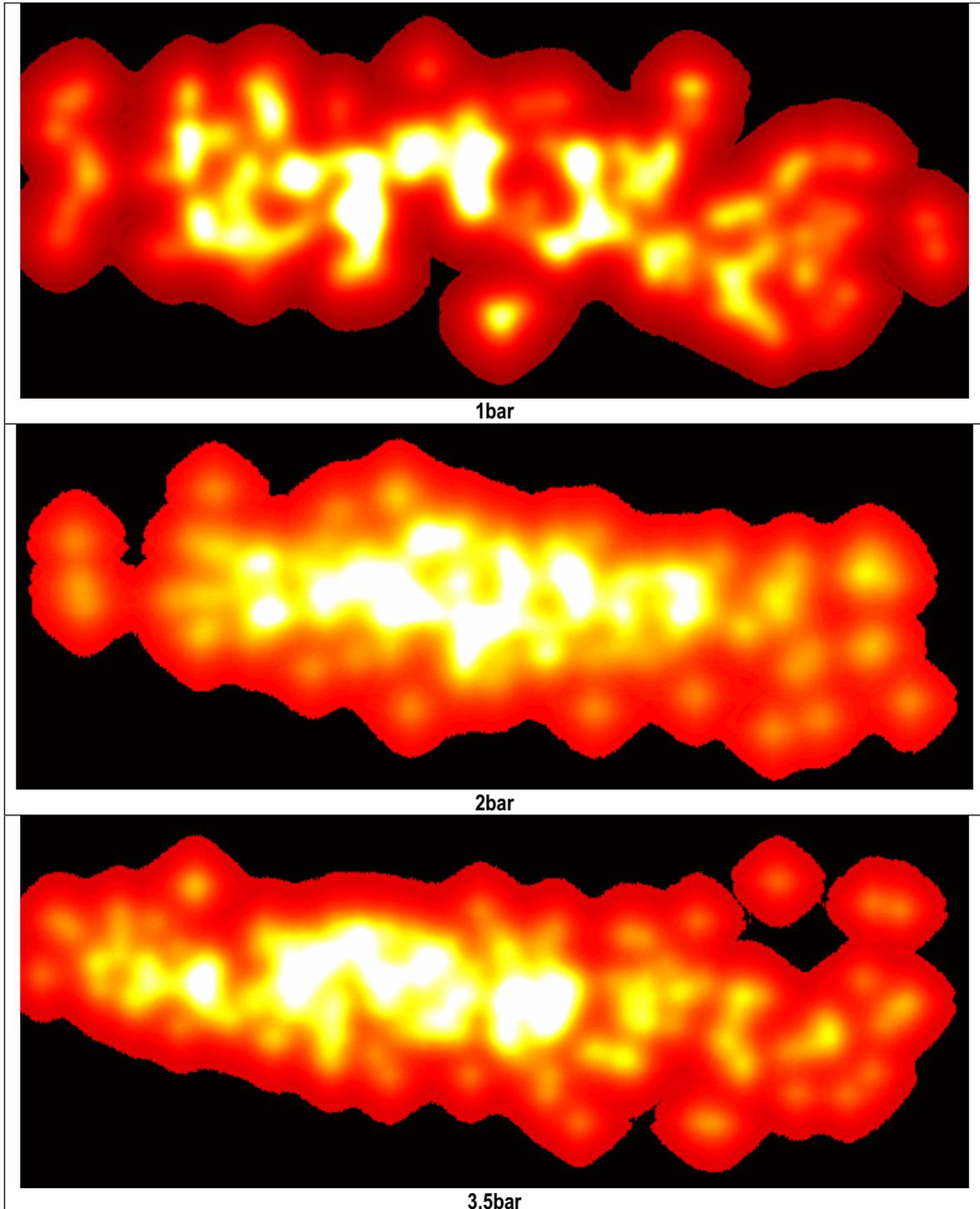
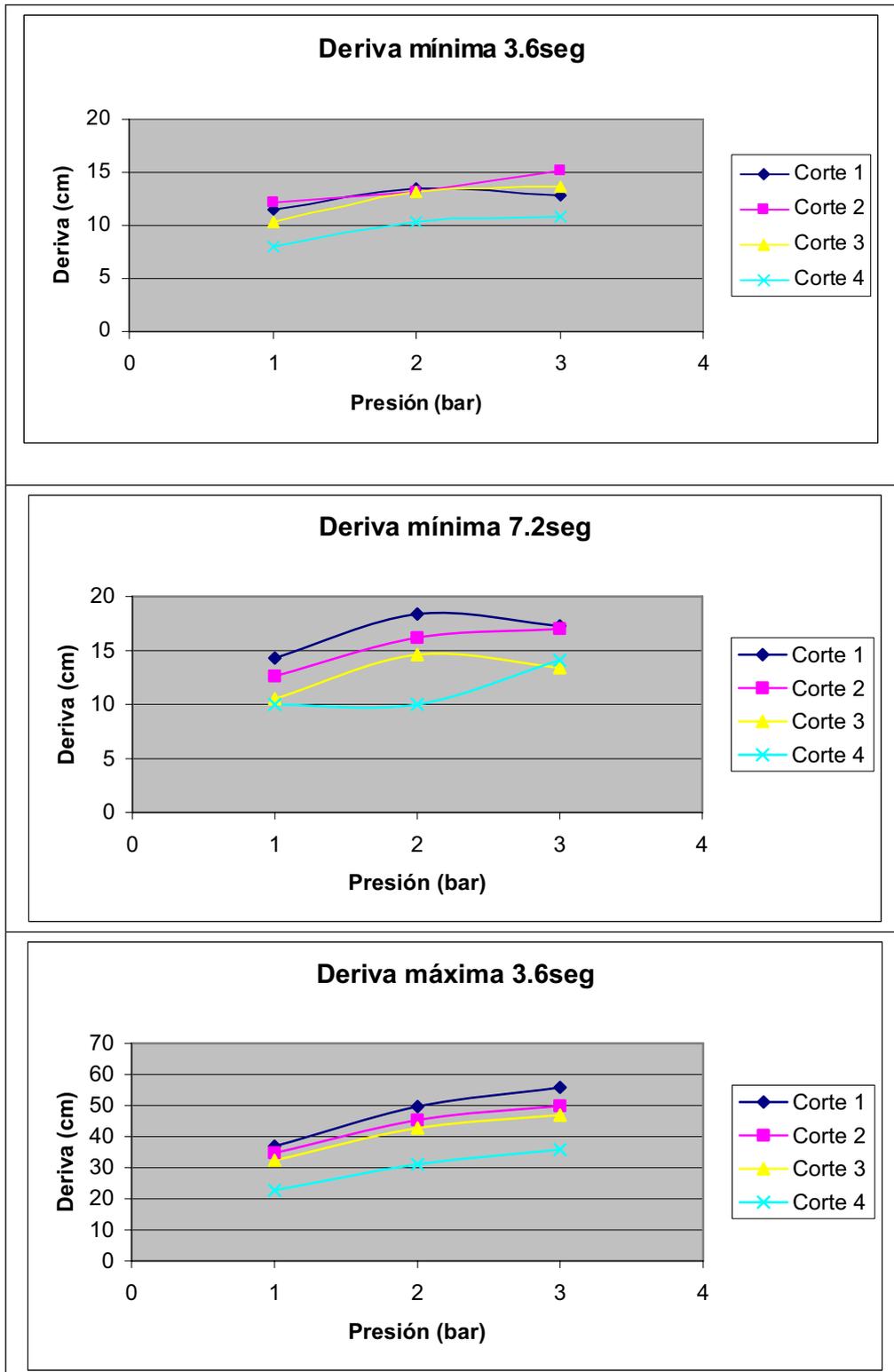


Figura 8. 8. Distribución de flujo en el corte inferior para distintas presiones.

8.1.3. Resultados en función del Tiempo de Promediado

Hasta ahora todas las medidas se han realizado con un tiempo de promediado de 3.6 segundos, sin embargo, el valor de la deriva depende de este tiempo de promediado como se puede apreciar en las gráficas de la tabla 8.2.



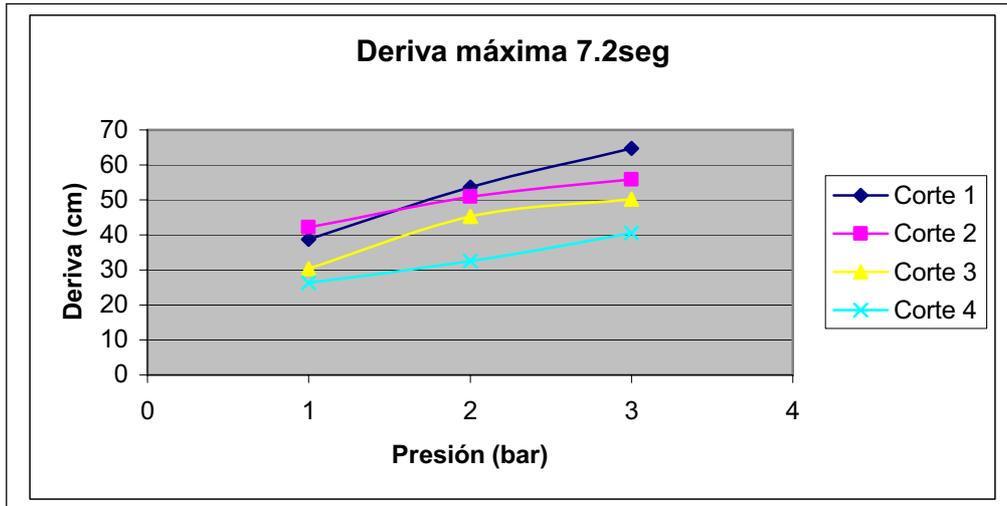
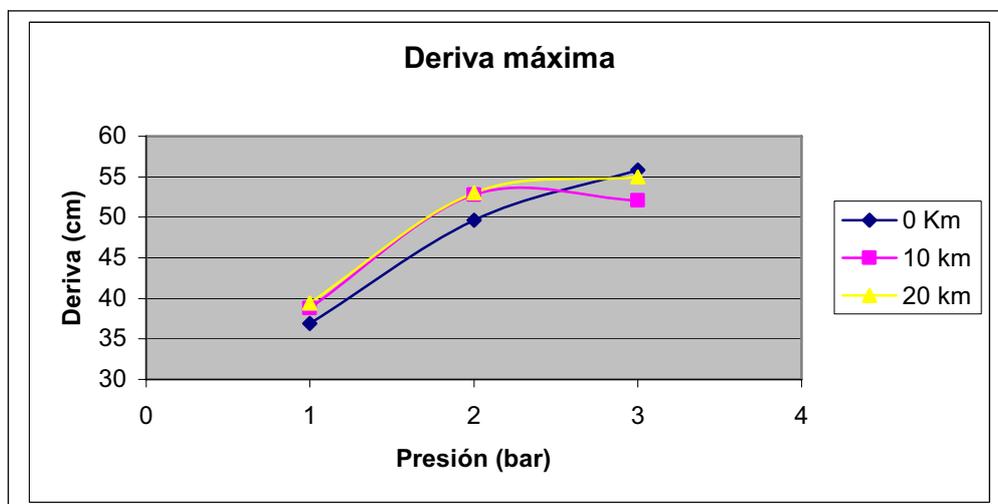


Tabla 8. 2. Comparativa de derivas para tiempos de promediado de 3.6 y 7.2 segundos.

En los resultados de la tabla 8.2 se observa como para tiempo de 7.2seg se obtienen unas derivas mayores, lo cual tiene sentido ya que al aumentar el tiempo, por probabilidad, aparecerán gotas cada vez más alejadas. Entonces, ¿cuál es el tiempo de promediado más adecuado?, la respuesta definitiva quizás deba fijarse mediante un estudio mucho más exhaustivo y dependiendo también de la aplicación que se le vaya a dar al pulverizador. Sin embargo, en este momento se puede afirmar que cuando la aplicación de visión realiza el recuento de las gotas, si el tiempo de promediado es muy alto, la aplicación puede saturarse generando una vista de distribución de flujo no representativa de la realidad. Del otro lado, si este tiempo es muy pequeño la aplicación capturará pocas gotas y no podrá realizar una estimación correcta de la distribución de flujo, además de que calculará una deriva no representativa.

8.1.4. Resultados en función del Viento

En presencia de viento se añade una nueva componente a la dirección de movimiento de las gotas que llevará la dirección del viento. Esta nueva componente, como es lógico, incrementará el valor de las derivas, sobre todo las mínimas cuya dirección es aproximadamente paralela a la dirección del viento.



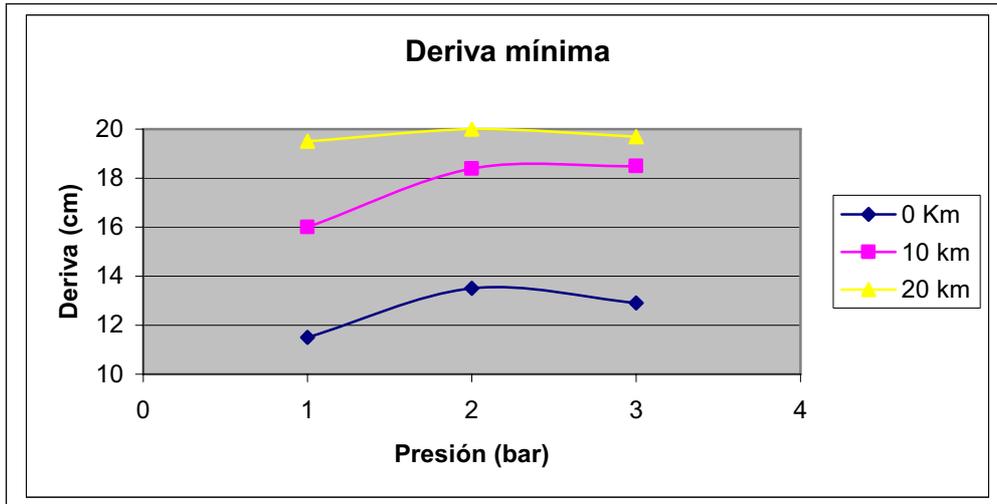
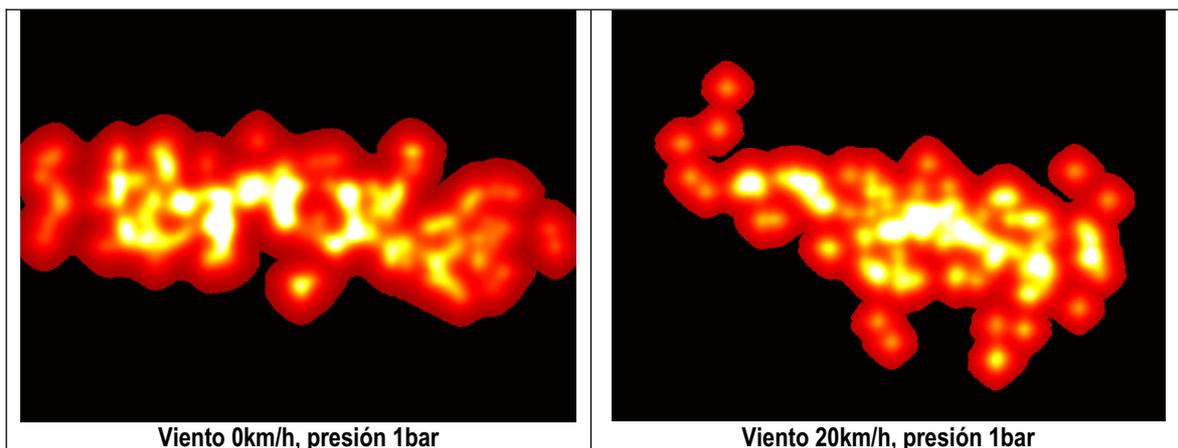


Tabla 8. 3. Derivas para el corte inferior en función del viento.

De los resultados de la tabla 8.3 se puede extraer conclusiones como que existe una dependencia directa con la velocidad del viento pero que en el caso de las derivas máximas (dirección aproximadamente perpendicular a la del viento) es menos importante que la dependencia con respecto a la presión, y en el caso de las mínimas la dependencia con el viento es más acusada. Para ambos casos la dependencia con el viento parece anularse al incrementarse la presión, éste es importante ya que AUTOMAN intentará trabajar siempre con la presión mínima que garantice una pulverización sin influencia de viento.

Desde la vista de flujo se puede apreciar las alteraciones en su distribución que implica la presencia de viento. En los resultados de la tabla 8.4 se observa como al introducir viento la región de corte se deforma de su forma aproximada de rectángulo que posee en ausencia de éste, además dentro la distribución dentro de la región se aleja aún más de la uniformidad. Estos efectos se ven atenuados si se incrementa la presión sobre el pulverizador.



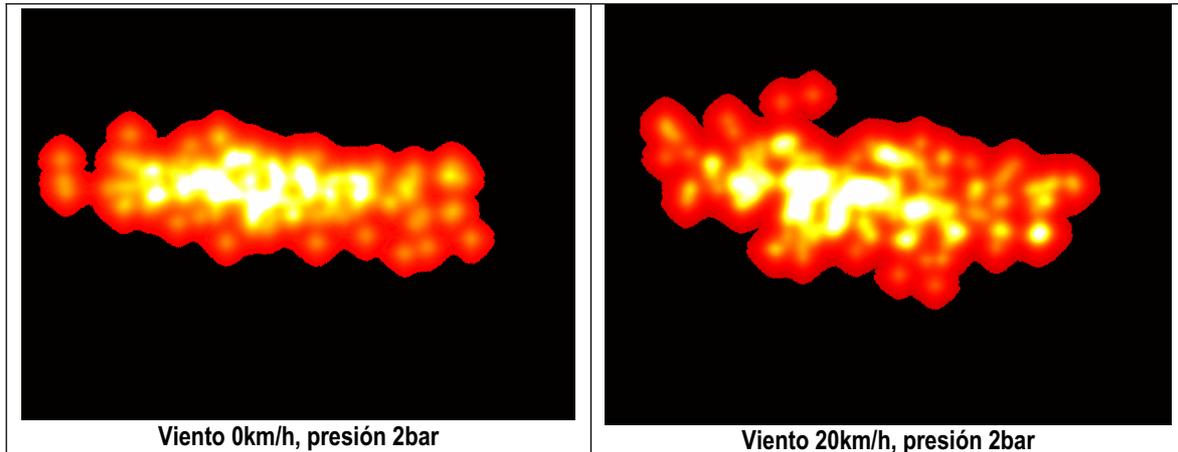


Tabla 8. 4. Variación de la distribución de flujo en relación al viento y la presión. En todas las imágenes la dirección de la componente principal de viento sería vertical y en sentido descendente.

Las pruebas con viento se pueden interpretar, suponiendo ausencia de viento en el entorno, como que el pulverizador va viajando sobre un vehículo, que podría ser el prototipo AUTOMAN, a la velocidad que marca el anemómetro. Entonces la vista de flujo, que muestra el promedio durante un determinado tiempo del flujo de gotas, se puede decir que muestra el flujo promedio sobre una distancia definida por el producto de la velocidad y el tiempo promedio.

Un pulverizador montado sobre el prototipo AUTOMAN se desplazará a la velocidad del prototipo, barriendo una determinada franja cuya longitud perpendicular a la dirección de movimiento la define la deriva máxima, y el ancho la deriva mínima. A través de la vista de flujo se puede conocer la distribución del barrido a lo largo de la dirección perpendicular del movimiento, para una distancia definida por el tiempo de promediado empleado y la velocidad, proyectando la imagen de la vista sobre una dimensión.

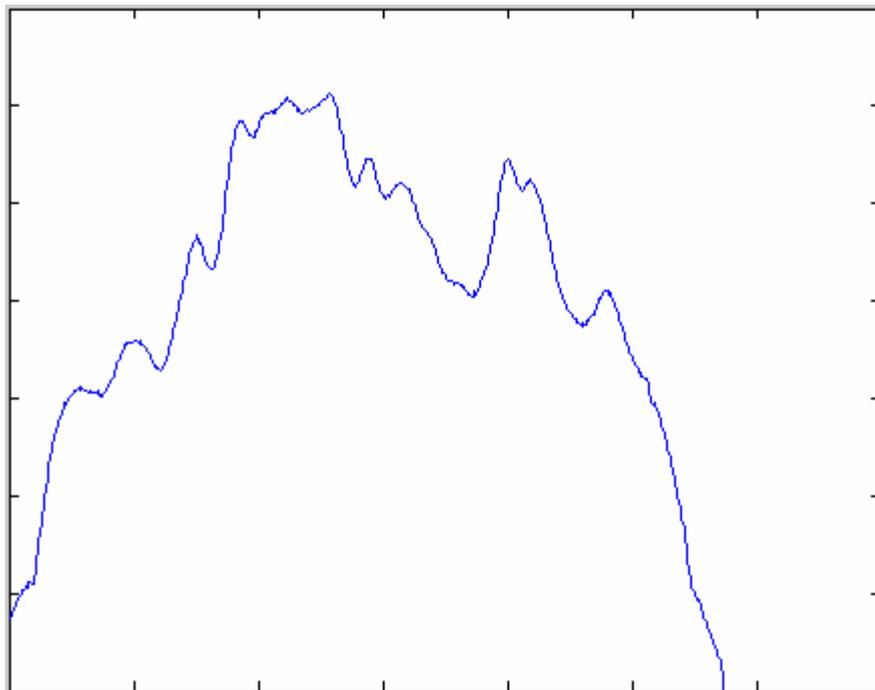


Tabla 8. 5. Distribución en dirección perpendicular en un barrido de 10 metros a 10Km/h con 2bar de presión.

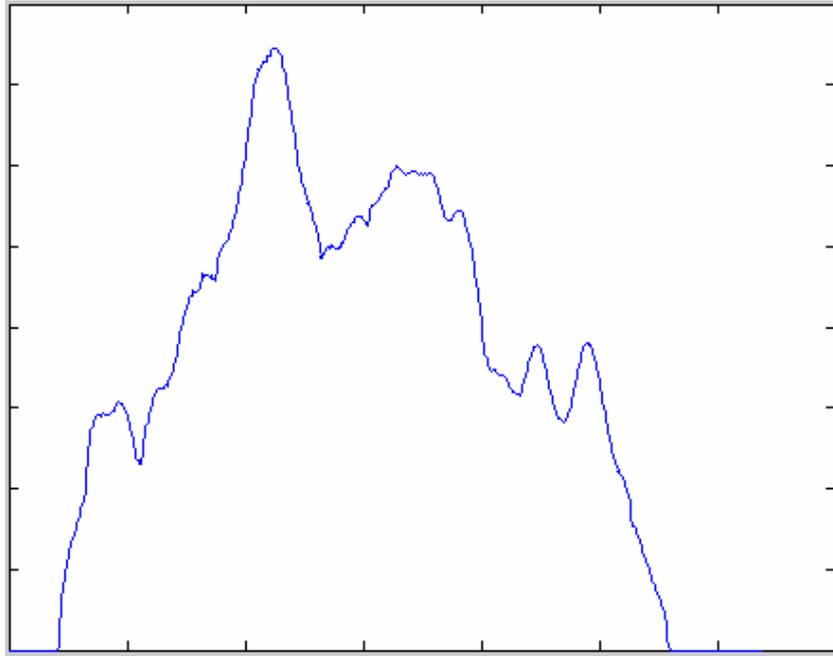


Tabla 8. 6. Distribución en dirección perpendicular de un barrido de 10 metros a 20km/h con 2bar de presión.

8.2. Aportaciones o Conclusiones del Proyecto

Entre las aportaciones realizadas por este Proyecto cabe destacar las siguientes:

- AUTOMAN-TUNEL constituye un banco de pruebas de laboratorio, que facilita la evaluación del comportamiento del prototipo AUTOMAN en campo. Ayudando, a través de los estudios que se realicen en este banco de pruebas, en el diseño del sistema de inyección (o pulverización) del prototipo, así como a la implementación del control del sistema de inyección de forma que éste sea autorregulable según las condiciones de viento del entorno.
- Sistema neumático de escaneado que permite hacer barridos con luz estructurada, haz plano, de manera que una aplicación de visión podría hacer una reconstrucción 3D del objeto que se escanea. En este Proyecto se limita solo ha cuatro cortes de la nube pulverizada, ya que son suficientes para las exigencias del proyecto, sin embargo este sistema permitiría una reconstrucción 3D de los objetos escaneados con una precisión de 1 o 2cm.
- Sistema de visión artificial capaz de medir las derivas sobre la nube de pulverización, límites máximos y mínimos de la región que ocupa la nube, a partir de la información generada por el sistema de escaneado a diferentes alturas.
- Además, el sistema de visión también es capaz de realizar una estimación aproximada del flujo de líquido dentro de la región de pulverización.
- El sistema de visión es capaz de trabajar con agua sin necesidad de adición de ningún colorante, lo que supone un alargamiento del periodo de vida de los componentes del sistema expuestos al paso del líquido, conductos, pulverizadores, cámara, sistema de iluminación, sensor de ultrasonidos y actuador neumático.

- Control automático del sistema de escaneado a través de un autómatas programable, el cual se conecta a un bus como maestro, al que a su vez se conectará el PC que actúa como procesador del sistema de visión y que actúa como esclavo. De manera que mediante un enlace PROFIBUS-DP ambos sistemas, visión y control, se integren en aras de alcanzar una automatización completa del sistema de escaneado. Solo faltaría por controlar desde el PC al sistema de ventilación y el sistema de inyección, que aún se realizan desde el cuadro eléctrico.
- Este control automático es el que permite poder ajustar el rango en el que se va a mover el sistema de escaneado, mediante un procedimiento sencillo y rápido sin que se tenga que modificar ni el hardware ni el software del sistema.
- La comunicación desde la aplicación de visión con el dispositivo de control del sistema de escaneado se realiza a través de un servidor OPC. Esta arquitectura de sistema de automatización ofrece varias ventajas frente al enfoque tradicional basado en el empleo de drivers, en concreto permitirá que la futura aplicación que controle el prototipo AUTOMAN y la aplicación de visión de AUTOMAN-TUNEL puedan ejecutarse simultáneamente desde el mismo PC, compartiendo el mismo dispositivo de control.
- La clase ServerOPC escrita en C++ y que actúa como cliente síncrono de un servidor OPC. Esta clase puede ser reutilizada por cualquier aplicación (escrita en C++) que desee conectarse a un servidor OPC y desee comunicarse con él, como por ejemplo pudiera ser la aplicación de control del prototipo AUTOMAN.
- En este Proyecto se ha adquirido gran experiencia en la configuración y programación del autómatas, y en uso y manejo de los sensores y actuadores de la compañía Festo®. La cual será muy útil cuando se emprenda la labor de implementar el sistema de control del prototipo AUTOMAN, ya que para esta tarea se empleará al mismo autómatas, los mismos sensores y actuadores similares.

8.3. Trabajos Futuros

El sistema desarrollado en este Proyecto no está ni mucho menos cerrado, sino que admite multitud de ampliaciones y mejoras, las cuales pueden clasificarse según su alcance en tres grupos: las que conciernen al sistema de control, las que conciernen al sistema de escaneado y las que conciernen al sistema de visión. La realización de todas ellas conllevaría tener un sistema en el que se pudiera generar la información suficiente, para poder realizar reconstrucciones 3D y 4D de la nube de pulverización con una precisión fijada de antemano, es decir el usuario en tiempo de ejecución podría decidir la cantidad de cortes.

En principio el sistema de control funciona de acuerdo a como a los objetivos iniciales que se plantearon. No obstante, para poder alcanzar un control centralizado desde el PC de todos los componentes del túnel se debería de realizar las siguientes mejoras:

- Controlar el sistema de ventilación y de inyección, que en este momento se realiza mediante el cuadro eléctrico del túnel. Esta tarea se presume sencilla, ya que sólo implicaría controlar los interruptores del cuadro con relés conectados al autómatas.
- Controlar la posición del pulverizador, ahora esta posición se fija manualmente. Esta mejora se presenta más complicada que la anterior, ya que necesitaría de dos nuevos

actuadores, uno para desplegar y replegar la tijera (donde se encuentra el pulverizador) y otro variar su altura. Esto, a parte de complicaciones mecánicas, supondría también un aumento de la complejidad del sistema de control, ya que estos actuadores necesitarían de sensores que controlase la posición.

Las mejoras que se sugieren para el sistema de escaneado, y que se enumeran a continuación, van principalmente orientadas a aumentar la precisión en proceso de escaneado:

- Utilizar un actuador basado en engranajes, y no en un cilindro neumático, los cuales ofrecen una mayor precisión en sus movimientos. De manera que con este tipo de actuadores la altura a la que se realizan las paradas sería más controlable, y por tanto más precisa. Además si este es alimentado eléctricamente el sistema se ahorraría también el uso del motor compresor de aire. El empleo del actuador neumático se justifica por su disposición en el laboratorio, y porque este Proyecto también ha servido un poco como banco de pruebas de los componentes del prototipo AUTOMAN.
- Mejorar el sistema de iluminación aumentando su intensidad, lo que ayudaría a detectar las gotas de muy pequeño tamaño. También sería interesante el aumentar la amplitud de su haz con el propósito de no limitar la deriva máxima que se puede medir en la dirección del viento a 53cm, lo que podría quedarse corto en pulverizadores que generen una nube muy expandida, o cuando el viento es muy alto. Una solución que no supondría elevar mucho el coste del sistema que se propone en esta memoria, sería utilizar un array de láseres de haz plano de bajo coste (como el que se emplea en estos momentos), cuyas dimensiones (cantidad y separación) habrán de estimarse.

En cuanto al sistema de visión se proponen una serie de mejoras orientadas a migrar el sistema AUTOMAN-TUNEL actual, limitado a realizar medidas de solo 4 cortes de la nube, a un sistema en el que la precisión con la que se genere la información pueda ser ajustada por el usuario, para ello sería necesario cambiar el sistema de calibración, apareciendo dos alternativas distintas:

- Añadir unos soportes desmontables al vástago del actuador donde se pudiera colocar la rejilla de calibración, de manera se pudiese realizar una calibración automática de todos los cortes 2D (suponiendo máxima precisión) programando las rutinas correspondientes al autómatas y al sistema de visión. Esta opción tiene el inconveniente de la rigidez que impone el que solo se podrán realizar cortes que previamente hayan sido calibrados, además necesitaría implementar unos algoritmos de preprocesado de imagen que eviten la generación de errores durante la llamada a los algoritmos de calibración.
- Realizar una calibración 3D, la cual es más imprecisa que la anterior pero mucho más flexible, y además simplificaría más el proceso de calibrado al no involucrar al sistema de control. Su principal inconveniente es que no está contemplada por ActiveMIL, lo que implicaría el desarrollo de los algoritmos de calibración, que en el caso 3D son mucho más complejos que en el 2D, desde el nivel más bajo.

Dentro del sistema de visión, y más concretamente en lo concerniente a la cámara, se proponen una serie de modificaciones que contribuirán a mejorar las prestaciones y/o aligerar la aplicación de visión:

- Utilizar una óptica con una menor distancia focal y con una mayor apertura (número F). Esto implicaría poder cambiar la posición de la cámara a una posición más cercana y

con un ángulo menor con respecto al plano de los objetos, esto redundaría en una mejor precisión al estar más cerca y porque se podría utilizar una calibración en la que se contemplasen las distorsiones. Al tener una menor distancia focal aumentaría la distorsión radial, sin embargo, esto no es problema porque ahora se pueden corregir con la calibración. Además, el hecho de tener una mayor apertura permitiría la entrada de más luz y mejorando las prestaciones de la aplicación.

- Emplear una cámara más sensible, o mejor dicho cuyo sensor presente una mejor relación S/N, de manera que se pueda aligerar el preprocesado de imágenes en la aplicación de visión artificial. Esto probablemente supondrá un aumento del coste del sistema.

APÉNDICE I

INCLUSIÓN DE LA CLASE ServerOPC EN VISUAL C++ .NET

Este apéndice se explica como utilizar la clase ServerOPC en aplicaciones que se desarrollan en el entorno Visual C++ .NET. La clase ServerOPC resuelve el problema de la comunicación con servidores OPC desde código C++, por lo que se puede aprovechar por otras aplicaciones de automatización escritas en este lenguaje de programación.

En este apéndice se parte de que se cumplen tres requisitos:

- Se dispone de Visual C++ .NET.
- Se tiene instalado en la máquina local, o en una remota conectada a la local, un servidor OPC.
- Tener el CD del este Proyecto con los archivos de definición e implementación de la clase ServerOPC (ServerOPC.h y ServerOPC.cpp) y el archivo IDL (OPC.idl).

Se parte de un proyecto ya creado y lo primero es copiar el archivo OPC.idl dentro del directorio del proyecto en una carpeta llamada "Include", si está no estuviera se crearía.

Se agrega una nueva carpeta en la vista del *Explorador de Soluciones*, después se agrega en esta carpeta el archivo IDL. En este archivo es donde se definen los interfaces y tipos de datos que ServerOPC va a poder utilizar.

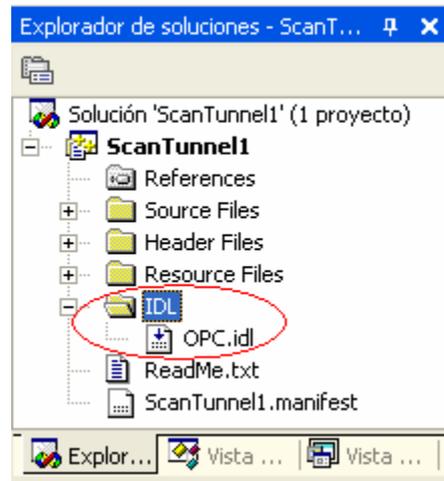


Figura I. 1. Vista del Explorador de Soluciones una vez añadida la carpeta y el archivo IDL.

Una vez se tiene añadido el archivo IDL, se pasa a ajustar la configuración del proyecto para que durante la compilación del proyecto también se contemple a este archivo y ésta se haga de forma correcta. El ajuste de la configuración se realiza desde *Propiedades del Proyecto* en el menú *Proyecto*, y lo primero es incluir (si no lo está ya) el directorio que contiene el archivo IDL (Include), y añadirle la definición `_WIN32_DCOM` al preprocesador (todo ello dentro de la carpeta C/C++).

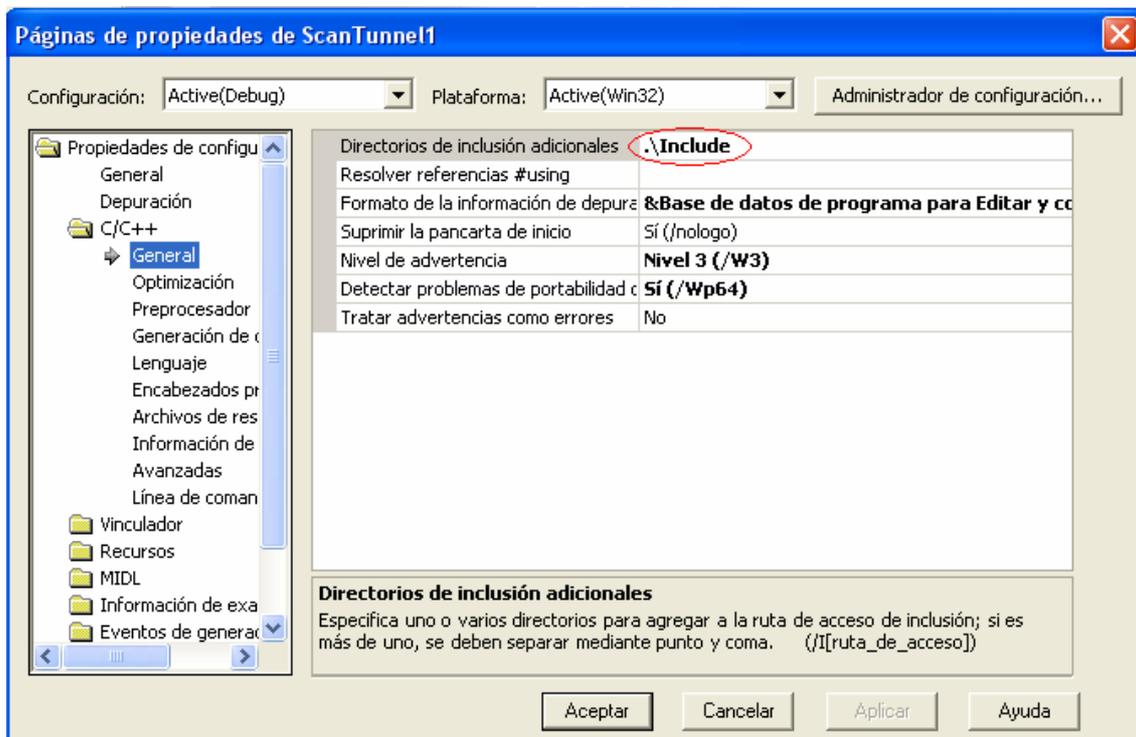


Figura I. 2. Inclusión del directorio con el archivo IDL.

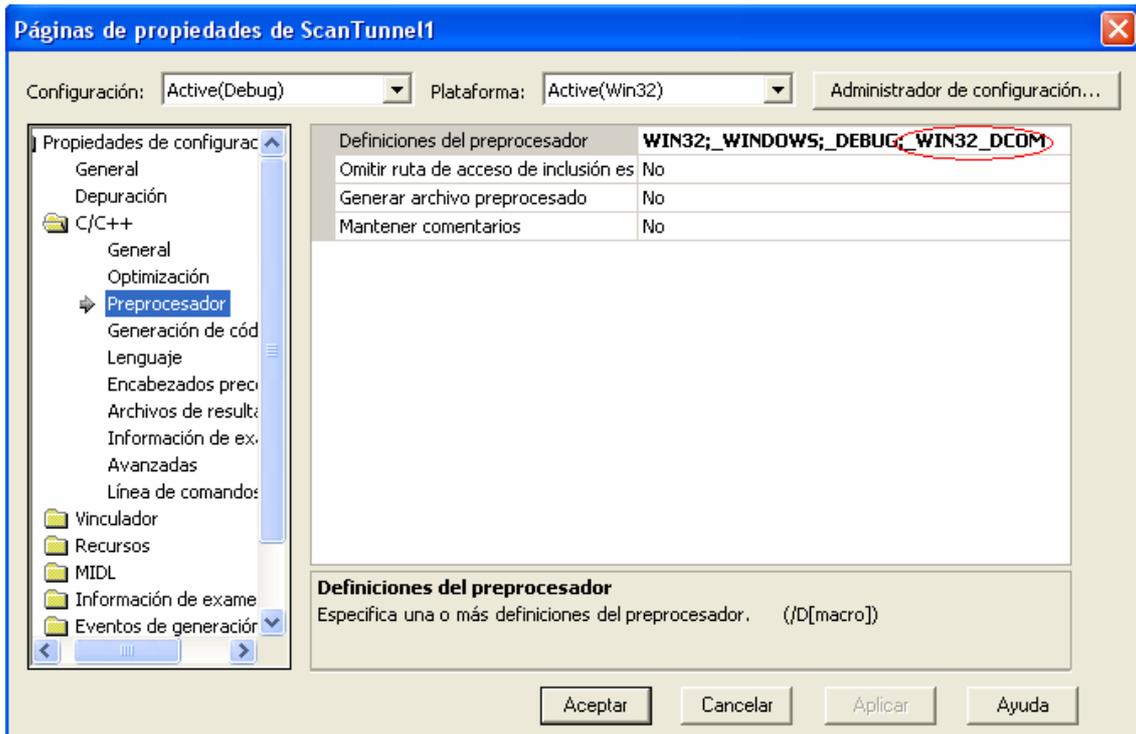


Figura I. 3. Adición de definiciones al preprocesador.

Ahora se configura al preprocesador MIDL (carpeta MIDL en *Propiedades de la configuración*), en donde de nuevo hay que especificarle el directorio que contiene el archivo IDL, decirle que genere un Proxy sin código auxiliar y definir el directorio donde se almacenarán los resultados de la compilación de OPC.idl. A partir del archivo IDL el compilador generará una serie de archivos en C (OPC_i.c, OPC_p.c, dlldata.c, y opc.h) que ServerOPC utilizará como *stub* en las llamadas a procedimiento remoto al servidor OPC.

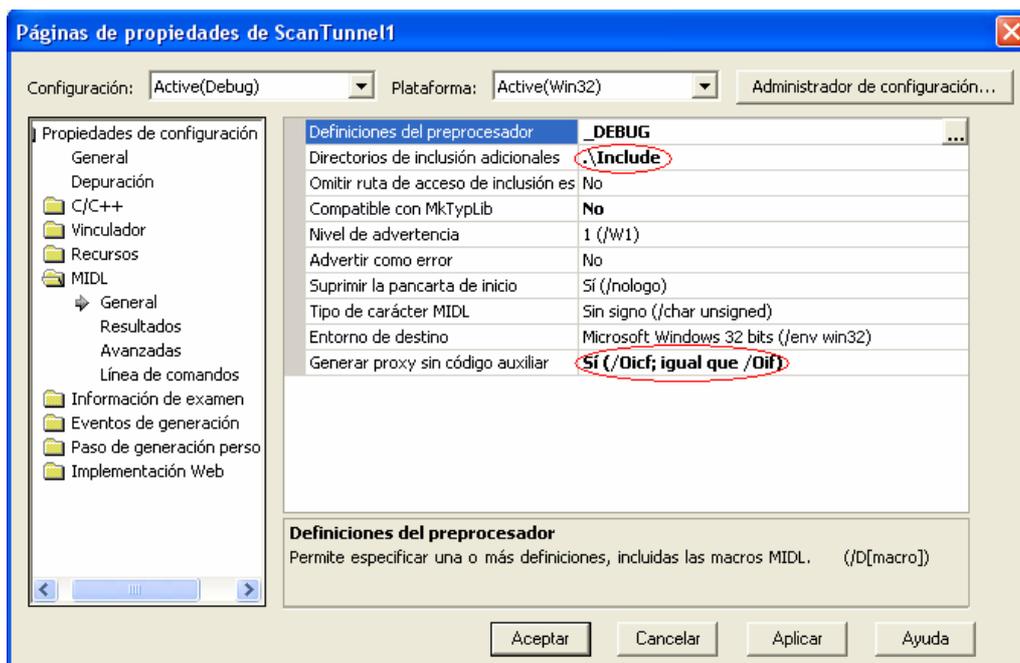


Figura I. 4. Configuración del preprocesador MIDL categoría General.

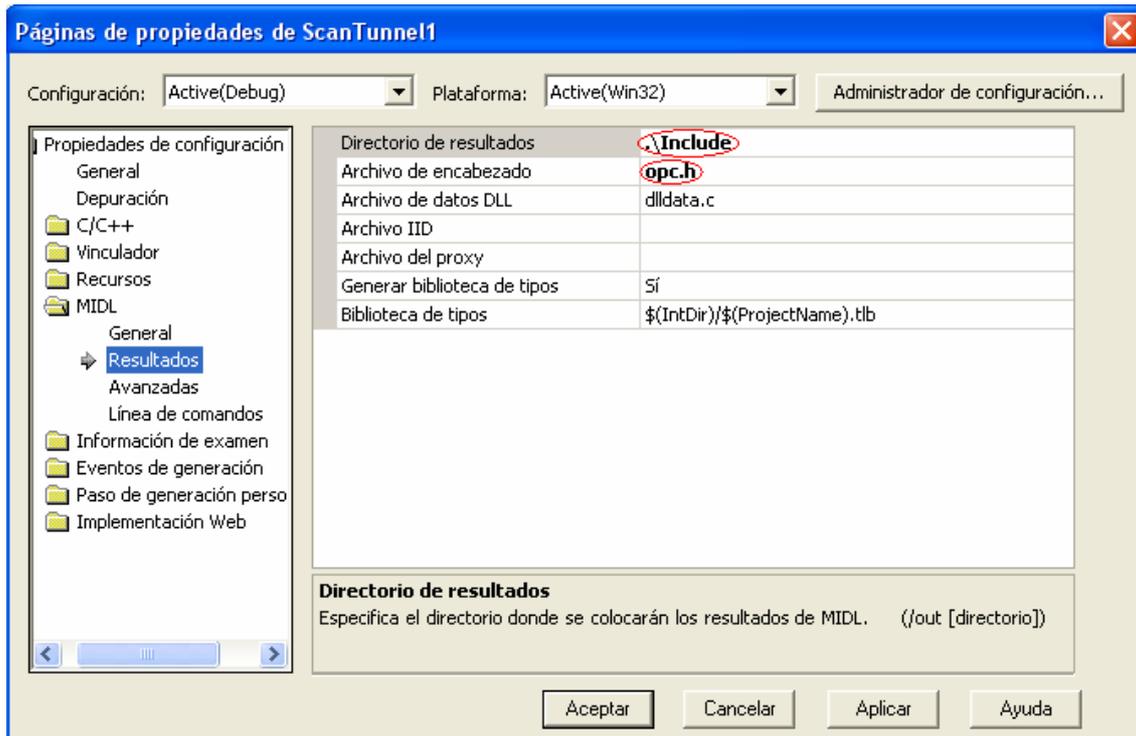


Figura I. 5. Configuración del preprocesador MIDL categoría *Resultados*.

Una vez el proyecto se ha configurado para que genere todos los archivos que la clase ServerOPC, se copian los archivos de la clase contenidos en el CD de este Proyecto (ServerOPC.h, ServerOPC.cpp) en el directorio del proyecto Visual C++, y se añaden al proyecto a través del Explorador de soluciones.

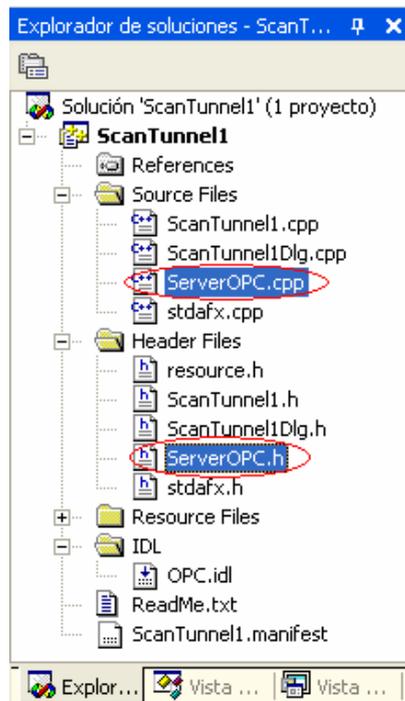


Figura I. 6. Inclusión en el proyecto de los archivos de ServerOPC.

Una vez realizados estos pasos para poder definir e instanciar objetos de la clase SeverOPC en cualquier archivo del código fuente de la aplicación, solo resta hacer uso de la directiva `#include` para incluir la definición del clase ServerOPC (`#include "ServerOPC.h"`).

A continuación se expone el interfaz de la clase ServerOPC:

Clase ServerOPC	
Archivos incluidos	
<code>opc.h</code>	Archivo que se obtiene de la compilación del fichero IDL y que contiene la definición de las clases e interfaces OPC en que se apoya ServerOPC
Constructor	
<code>ServerOPC(LPWSTR servName);</code>	<code>servName</code> Nombre del servidor OPE al que se quiere conectar
Métodos	
<code>void InstantiateServer();</code>	Inicialización, conecta con el servidor OPC
<code>void AddTheGroup(IOPCItemMgt* &pIOPCItemMgt);</code>	Añade un grupo en el servidor OPC, un grupo contiene una serie ítems cuyo valor se corresponde con variables de dispositivos conectados al servidor OPC
<code>pIOPCItemMgt</code>	Interfaz manejador de grupo sobre el que se devuelve el resultado
<code>OPCITEMRESULT* AddTheItems(LPWSTR itemID[], IOPCItemMgt* pIOPCItemMgt, const int numItems);</code>	Añade ítems al grupo y devuelve un interfaz a los conjunto de ítems
<code>itemID[]</code>	Array con los identificadores de los ítems
<code>pIOPCItemMgt</code>	Manejador de grupo sobre el que se añaden los ítems
<code>numItems</code>	Número de ítems
<code>void ReadItem(IOPCItemMgt* pIOPCItemMgt, OPCITEMRESULT* pItemResult, VARIANT& varValue, const int posItem);</code>	Lee un ítem del servidor OPC
<code>pIOPCItemMgt</code>	Manejador de grupo
<code>pItemResult</code>	Contenedor de ítems
<code>varValue</code>	Valor leído
<code>posItem</code>	Ítem que se desea leer
<code>void WriteItem(IOPCItemMgt* pIOPCItemMgt, OPCITEMRESULT* pItemResult, VARIANT varValue, const int posItem);</code>	Escribe en un ítem del servidor OPC
<code>pIOPCItemMgt</code>	Manejador de grupo
<code>pItemResult</code>	Contenedor de ítems
<code>varValue</code>	Valor escrito
<code>posItem</code>	Ítem que se desea escribir
<code>void RemoveItems(IOPCItemMgt* pIOPCItemMgt, OPCITEMRESULT* pAddResult, const int numItems);</code>	Elimina los ítems del grupo
<code>pIOPCItemMgt</code>	Manejador de grupo
<code>pItemResult</code>	Contenedor de ítems
<code>numItems</code>	Número de ítems
<code>void RemoveGroup();</code>	Elimina el grupo añadido

```
void RemoveGroup ();  
Desconecta con el servidor OPC
```

Tabla I. 1. Interfaz de la clase ServerOPC.

APÉNDICE II

ActiveMIL EN VISUAL C++ .NET

En este apéndice se explica como usar ActiveMIL, incluir controles y trabajar con los objetos que los representan en Visual C++ .NET. La existencia de este apéndice se explica por que la guía de usuario de ActiveMIL 8.0 [8] explica como hacerlo en Visual Basic .NET pero no en Visual C++ .NET, luego este apéndice es interesante para gente que comience a trabajar con esta librería.

Este apéndice se basa en el desarrollo de un sencillo ejemplo que parte de los siguientes requisitos:

- Se tiene instalado Visual C++ .NET.
- Está instalado el CD MIL 8.0 y se dispone de licencia de desarrollo.
- Se ha instalado el Add-In de ActiveMIL y han añadido las rutas de archivos de inclusión y de librería a Visual C++ .NET [8].

Si se cumplen estos requisitos ya se poseen todas las herramientas necesarias para realizar el ejemplo. Se comienza creando un nuevo proyecto a partir de la plantilla que ofrece ActiveMIL.

Esta plantilla de inicio crea una aplicación MFC que contiene un diálogo que tiene insertados varios controles ActiveMIL:

- Application Control
- System Control
- Image Control
- GraphicContext Control

- Display Control

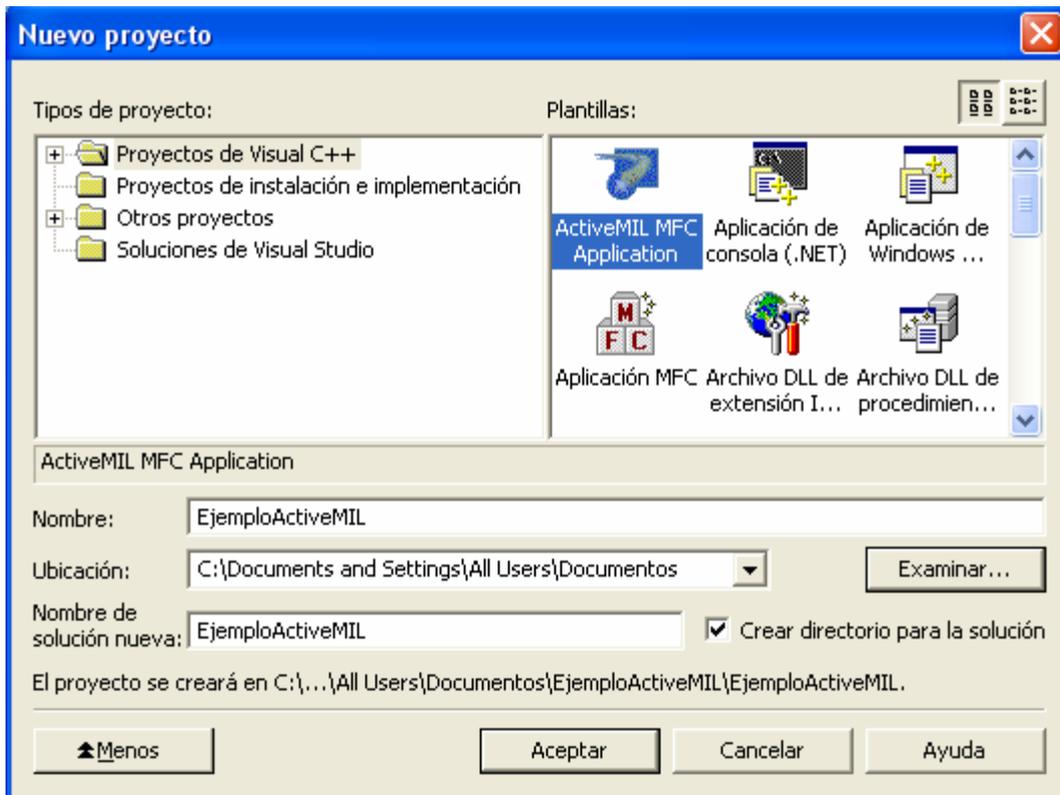


Figura II. 1. El ejemplo parte de la plantilla de ActiveMIL.

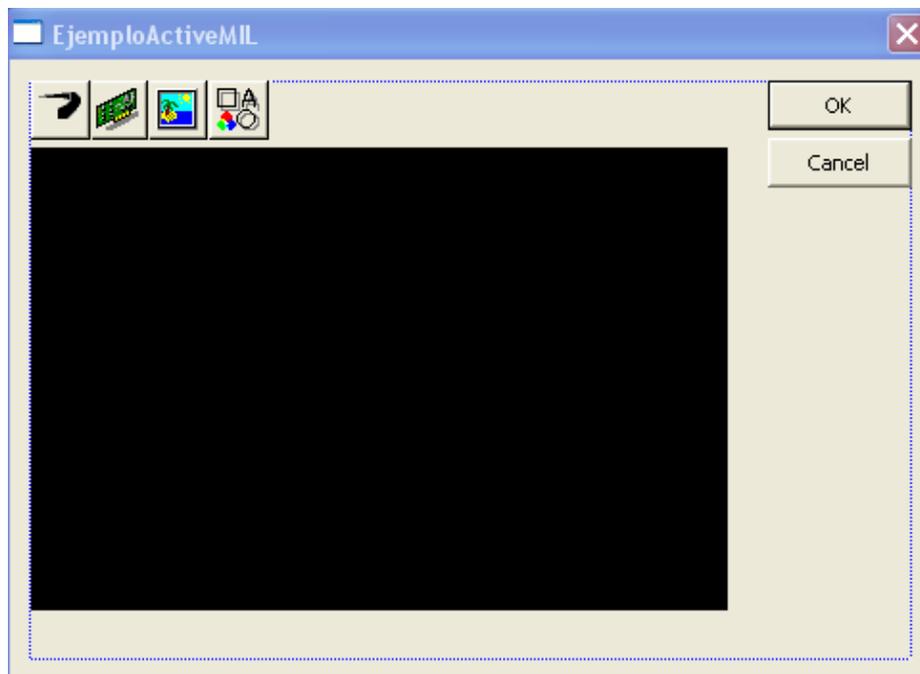


Figura II. 2. Interfaz con los controles ActiveMIL de la plantilla.

Estos controles ya pueden ser utilizados desde el editor de recursos, y desde el código (EjemploActiveMILDlg.cpp) de la aplicación utilizando los objetos declarados en el archivo de

definición de clase (EjemploActiveMILDlg.h), en el que previamente se incluyeron las librerías de estos controles. Es posible que si se dispone de hardware de adquisición y de una cámara también aparezca al comienzo, además de los controles mostrados en el ejemplo, un *Digitizer Control*.

```
#include "Application.h"
#include "System.h"
#include "Image.h"
#include "Display.h"
#include "GraphicContext.h"
```

Figura II. 3. Inclusión de archivos de cabecera para los controles ActiveMIL en EjemploActiveMILDlg.h.

```
CApplication m_Application1;
CSystem m_System1;
CImage m_Image1;
CDisplay m_Display1;
CGraphicContext m_GraphicContext1;
```

Figura II. 4. Declaración de objetos que representan a los controles ActiveMIL en EjemploActiveMILDlg.h.

Desde el archivo de implementación de la clase EjemploActiveMILDlg.cpp ya se puede llamar a los métodos de estos controles, por ejemplo, si se quiere abrir una imagen (esta ha de ser monocromo para que el ejemplo funcione) cuando se pulse OK en el diálogo de la aplicación basta con añadir el siguiente código:

```
void CEjemploActiveMILDlg::OnBnClickedOk()
{
    // TODO: Agregue aquí su código de controlador de notificación de control
    m_Image1->Load( "", true );
}
```

Figura II. 5. Código (en azul) a añadir para abrir una imagen en el que se utiliza uno de los controles con los que parte la plantilla ActiveMIL.

Ahora se desea hacer uso de la funcionalidad de un control que de partida no incorpora la plantilla. La inclusión de un nuevo control y utilizar su funcionalidad implica un mayor número de pasos, que comienzan con insertar el control en el diálogo en el editor de recursos. A modo de ejemplo se insertará un *ImageProcessing Control*.

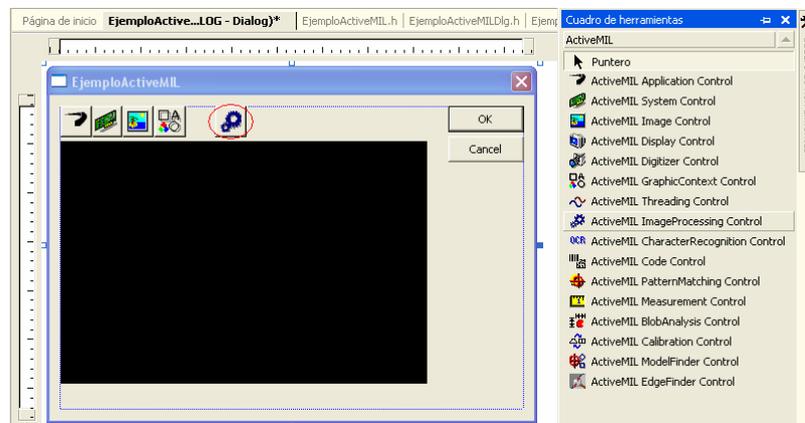


Figura II. 6. Adición del control ImageProcessing sobre el diálogo.

Desde el editor de recursos se pueden ya modificar las propiedades del control, sin embargo, aún no se puede utilizar desde el código de la aplicación. Por lo tanto, lo primero es incluir la librería del control y declarar un objeto que lo represente en el archivo EjemploActiveMILDlg.h.

```
#include "ImageProcessing.h"
...
...
CImageProcessing m_ImageProcessing1;
```

Figura II. 7. La línea superior contiene la inclusión del archivo de cabecera del controlador y la inferior la declaración del objeto que representará al control.

Pero el objeto declarado aún no representa al control incluido en el diálogo, para que esto suceda es necesario establecer un vínculo entre ambos para que haya una sincronización de los datos entre el control y el objeto que lo va a representar. Para establecer el vínculo lo primero es conocer el identificador del control, lo cual se puede hacer seleccionando el control desde la página de propiedades. El vínculo ha de establecerse en la definición del método que la clase EjemploActiveMILDlg dedica a ello.

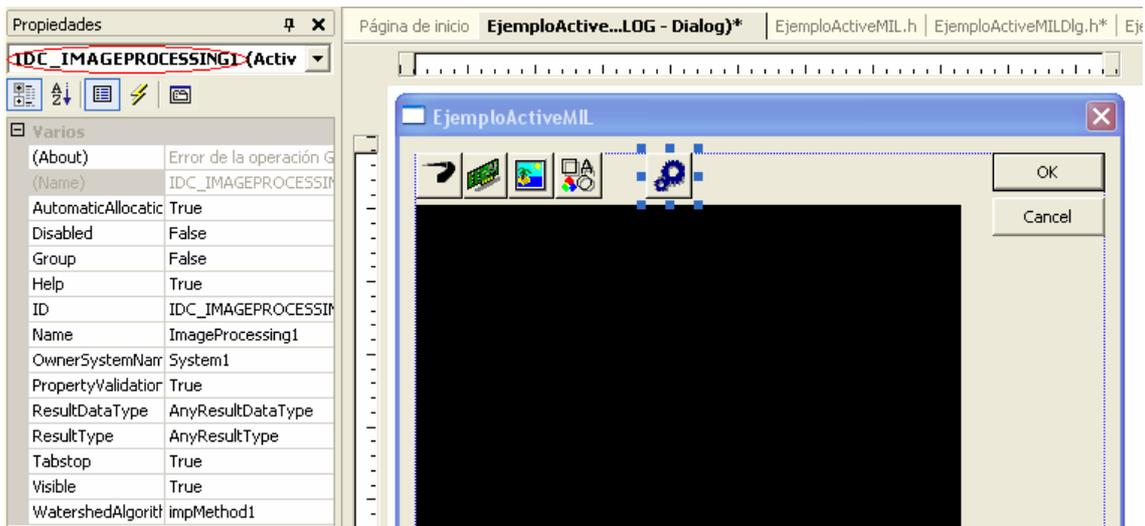


Figura II. 8. Identificador del control ImageProcessing.

```
void CEjemploActiveMILDlg::DoDataExchange (CDataExchange* pDX)
{
    CDialog::DoDataExchange (pDX) ;

    DDX_Control(pDX, IDC_APPLICATION1, m_Application1);
    DDX_Control(pDX, IDC_SYSTEM1, m_System1);
    DDX_Control(pDX, IDC_IMAGE1, m_Image1);
    DDX_Control(pDX, IDC_DISPLAY1, m_Display1);
    DDX_Control(pDX, IDC_GRAPHICCONTEXT1, m_GraphicContext1);
    DDX_Control(pDX, IDC_IMAGEPROCESSING1, m_ImageProcessing1);
}
```

Figura II. 9. Línea de código que establece el vínculo entre el control y el objeto que lo representa.

Ya se puede aprovechar la funcionalidad desde el código C++ empleando el objeto que representa el controlador. Para demostrar ésto se pueden escribir las siguientes líneas de código en la función que es llamada cuando se pulsa OK:

```
void CEjemploActiveMILDlg::OnBnClickedOk()
{
    // TODO: Agregue aquí su código de controlador de notificación de control
    m_Image1->Load( "", true );

    m_ImageProcessing1->Source1 = m_Image1;
    m_ImageProcessing1->Destination1 = m_Image1;
    m_ImageProcessing1->Binarize( impGreaterThan, amilDefault, 255 );
}
```

Figura II. 10. Código que a partir de la funcionalidad del control *ImageProcessing* binariza la imagen.

Una vez finalizado el ejemplo se puede generar y ejecutar para verificar su funcionamiento.

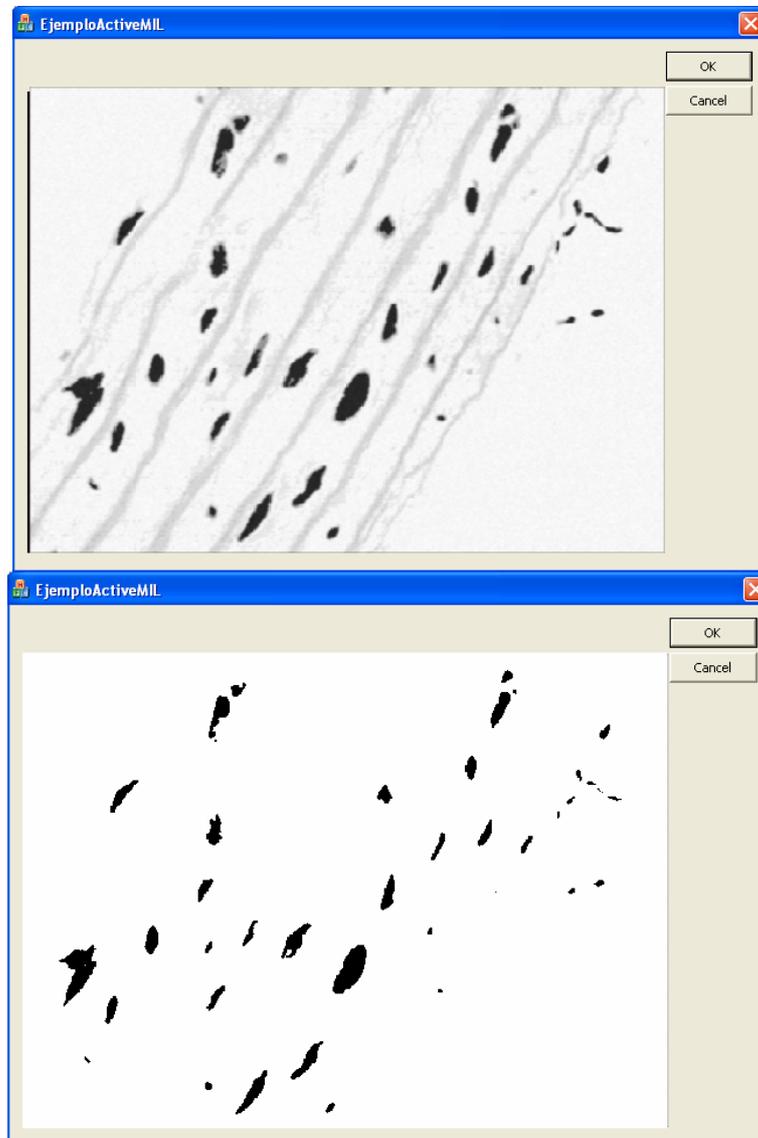


Figura II. 11. Funcionamiento de la aplicación del ejemplo.

BIBLIOGRAFÍA

- [1] Michael Moco, Edgar Neitt. "*Evaluating road maintenance equipment*", WFCOS, Porto, Portugal, Septiembre, 2000.
- [2] De la Escalera, A., "*Visión por Computador*", Prentice-Hall, 2001.
- [3] Martínez Sánchez, A., "*LADA-VISIÓN: desarrollo de un línea automatizada de descorazonado de alcachofas empleando técnicas de Visión Artificial*", Proyecto Final de Carrera, ETSIT, Universidad Politécnica de Cartagena, 2004.
- [4] J. González Jiménez, "*Visión por Computador*", Editorial Paraninfo, 2000.
- [5] G. Medioni, S. B. Kang, "Emerging Topics in Computer Vision", IMSC Press Multimedia Series, Prentice Hall, 2005.
- [6] R. Jain, B. Kasturi, B. G. Schunck, "Machine Vision", McGraw-Hill, 1995.
- [7] "Matrox Imaging Library 8, User Guide", Matrox Electronics Systems Ltd., 2005.
- [8] "Matrox ActiveMIL 8, User Guide", Matrox Electronics Systems Ltd., 2005.
- [9] Balcells J., Romeral J. L., "*Autómatas Programables*", Serie Mundo Eléctrico, Marcombo Boixareu Editores, 1997.
- [10] Mayol, Albert, "*Autómatas Programables*", Colección Productiva, núm. 3. Marcombo Boixareu Editores, 1987.

- [11] Hecht, Zajac, “*Óptica*”, Addison Wesley Iberoamericana, 1986.
- [12] Castillo, Juan Del, “*Tendencias en Arquitecturas de Control: OLE for Process Control*”.
- [13] Deitel, H. M., “*Cómo programar en C++*”, Pearson Educación, 1995.
- [14] Martínez Torres, J., “*STEP 7 Programación Avanzada*”, Edición 1.0, 1999.
- [15] The Math Works Inc., “*MATLAB Image Processing Toolbox, User’s Guide*”, Versión 2, 1997.
- [16] “*Matrox Intellicam: User Guide*”, Matrox Electronics Systems Ltd., versión 2.07, 2000.

Recursos Web

- [1] www.basler-vc.com, Web Basler Vision Technologies.
- [2] www.micron.com, Web Micron Technologies Inc.
- [3] www.wikipedia.org, Enciclopedia Libre Wikipedia.
- [4] www.matrox.com/imaging, Web de la compañía Martox Electronic Systems Ltd. dedicada a sus productos de visión artificial.
- [5] www.siemens.com, Web de la compañía Siemens Energy & Automation Inc.
- [6] www.profibus.com, Web de la organización PROFIBUS Internacional.
- [7] www.festo.com, Web del fabricante FESTO.
- [8] www.honeywell.com/sensing, Web sobre sensores de Honeywell International Inc.
- [9] www.schiltknecht.com, Web del fabricante de sensores Schiltknecht Messtechnik AG.
- [10] www.microsoft.com, Web de la compañía Microsoft Corporation, desde donde se puede tener acceso a la librería de documentación MSDN.
- [11] www.mathworks.com, Web de The Math Works Inc. compañía desarrolladora de Matlab®, y contiene información sobre procesado de imágenes.