

## 1. Introducción

Hoy en día es común encontrar cámaras de video en bancos, centros comerciales, escuelas, instituciones de gobierno, entre otros. Estas cámaras son utilizadas para vigilar las actividades que dan lugar en dichos lugares.



Figura 1. Uso de cámaras de video en varios lugares

Datos enviados al archivo o reutilización de los medios de almacenamiento

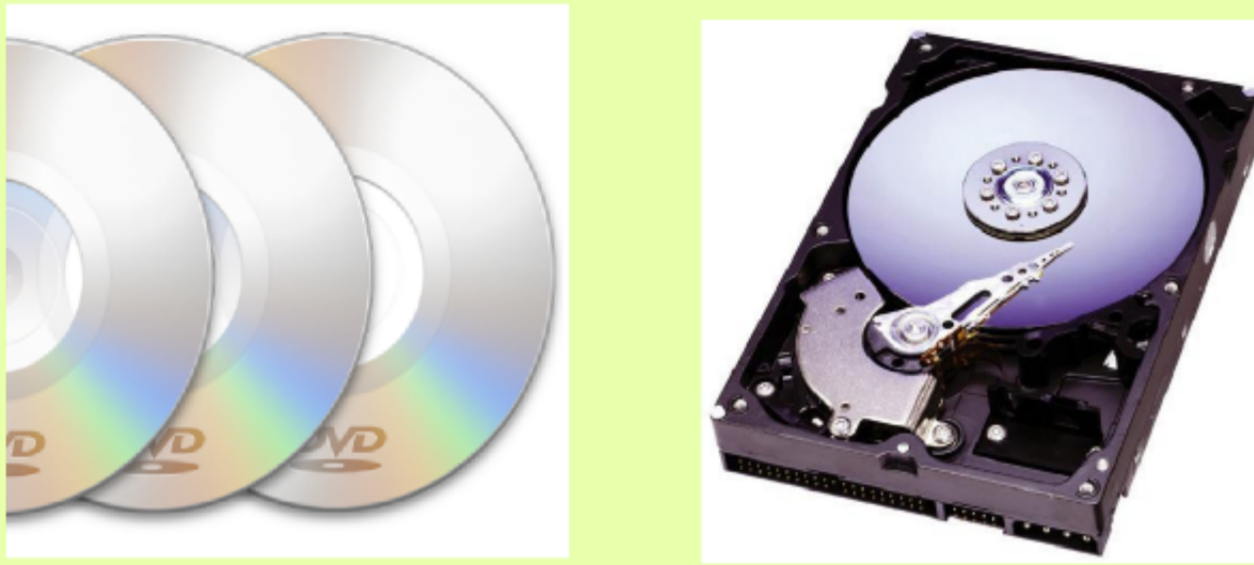


Figura 2. Almacenamiento de la información

En caso de ocurrir algún delito es entonces cuando se analiza la información, sin embargo, para entonces es demasiado tarde.



## 2. Problema

Es necesario un monitoreo constante de las imágenes capturadas por las cámaras de video.

Entre las posibles soluciones se encuentra la contratación de personal para encargarse de la supervisión y monitoreo de las cámaras de video.



Figura 3. Supervisión y monitoreo de actividades

Sin embargo, debido al gran número de cámaras a supervisar y al cansancio del cuerpo humano, no todas las actividades son observadas y reportadas oportunamente.

## 3. Trabajo relacionado

VSAM	VSIP	General Method
<ul style="list-style-type: none"> <li>+ Fusión de sensores</li> <li>+ Ambientes externos</li> <li>+ Redes neuronales para reconocimiento de actividades</li> <li>+ Reporte de actividades enviado a un servidor web</li> <li>+ Detección de movimientos, seguimiento y clasificación</li> </ul>	<ul style="list-style-type: none"> <li>+ Una cámara</li> <li>+ Ambientes internos</li> <li>+ Conocimiento a priori de la escena y las áreas de interés</li> <li>+ Uso de árboles AND/OR para reconocimiento de comportamientos</li> </ul>	<ul style="list-style-type: none"> <li>+ Una cámara</li> <li>+ Ambientes externos</li> <li>+ Conocimiento a priori de la escena y las áreas de interés</li> <li>+ Reporte de actividades en formato texto</li> <li>+ Generación de reportes a partir de reglas</li> </ul>

Tabla 1. Trabajo relacionado

## 4. Solución propuesta

Automatización del proceso de monitoreo de las cámaras de video utilizando técnicas de procesamiento y análisis de imágenes digitales.

- Reconocimiento de actividades para la toma de decisiones.

- Un solo operador para el monitoreo de áreas grandes al poner atención únicamente a las actividades sospechosas.

## 4. Solución propuesta cont ...

### 4.1 Módulo de visión



Secuencia de imágenes en disco o capturadas por una cámara



Detección de movimiento

- Diferencia temporal
- Fondo adaptable
- Análisis temporal de píxeles
- Eliminación de ruido
- Generación de regiones

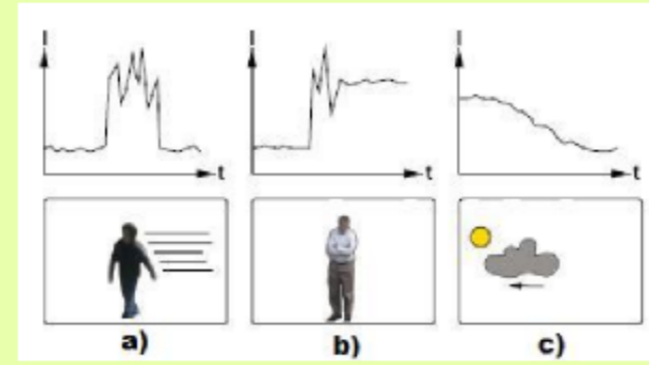


Figura 4. Análisis temporal de píxeles



Seguimiento de objetos

$$C(r) = \sum_{x \in R} \frac{W(x) |I_n(x) - I_{n+1}(x)|}{\|W\|}$$

- Medida de permanencia para eliminar falsas alarmas

### 4.2 Módulo de interpretación



Ancho, alto, área, circularidad, relación de aspecto

Clasificación de objetos

- Clases identificadas
- Personas
  - Autos

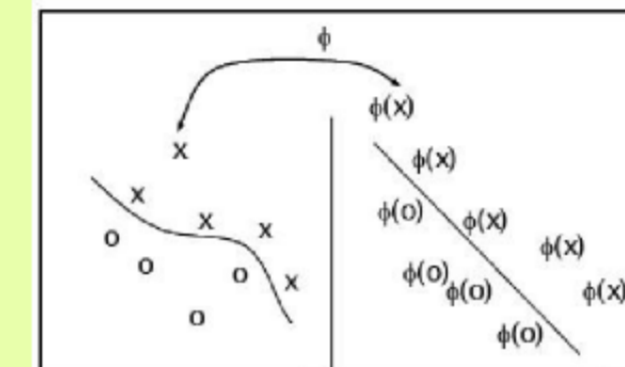


Figura 5. Mapeo de características con SVM



Reconocimiento de comportamientos

- Comportamientos primitivos
- Detenido
  - En movimiento
  - + Caminando o corriendo

### Comportamientos complejos

- Modelos Ocultos de Markov
- Cada comportamiento se modela como una secuencia de observaciones



Figura 6. Comportamiento "Ir a desayunar"

### Comportamientos detectados

- Personas platicando
- Persona atropellada
- Robo a personas

## 5. Desarrollo de software

Debido a la modularidad del método presentado, se optó por diseñarlo e implementarlo bajo el paradigma de programación orientado a objetos.

Con este enfoque se esperaba obtener un sistema flexible, adaptable al cambio y de fácil mantenimiento. Un análisis posterior reveló que estas características no fueron alcanzadas.

La creación de sistemas con flexibilidad al cambio es una tarea exclusiva para desarrolladores experimentados, Gamma *et al.* [4].

### 5.1 ¿Cómo diseñan los desarrolladores expertos?

- Cada vez que se encuentra una *buena* solución, se utiliza una y otra vez.

- En estas soluciones se encuentran patrones de clases y/o comunicación entre objetos.

## 5. Desarrollo de software cont ...

### 5.2 De la Arquitectura de las construcciones al diseño de sistemas de software

- En los 70's, el Arquitecto Christopher Alexander establece el lenguaje de patrones en [5].

- En los 90's, se aplican las ideas de Alexander en el diseño de sistemas de software.

### 5.3 La perspectiva de Shalloway et al. [6]

Conceptual	Conceptos en el dominio del problema
Especificación	Software a nivel de interfaces
Implementación	Código

Tabla 2. Perspectivas en el diseño orientado a objetos

## 6. Arquitectura de software

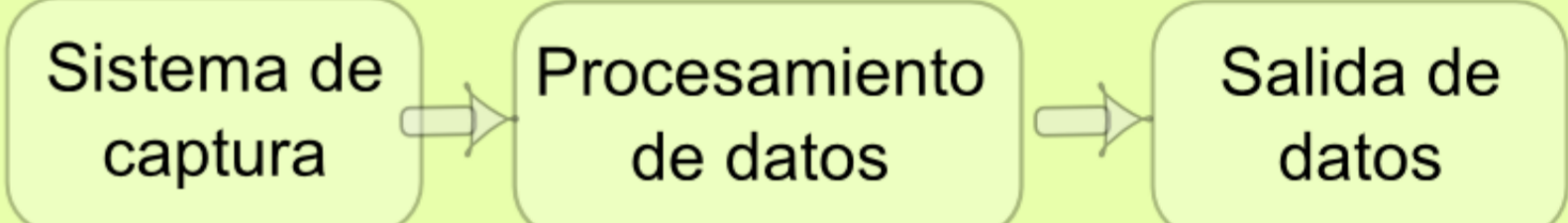


Figura 7. Arquitectura general

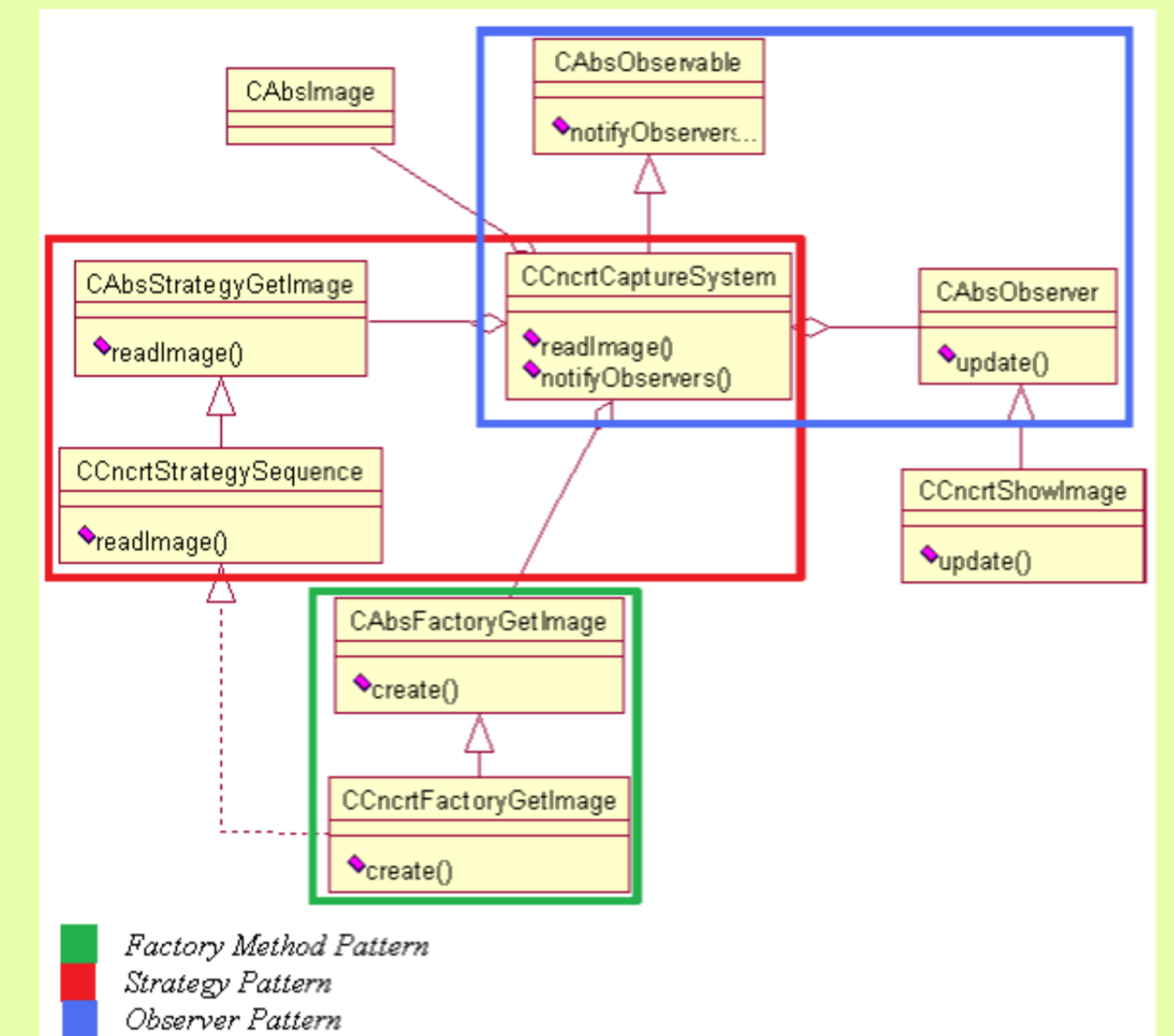


Figura 7. Sistema de captura y salida de datos

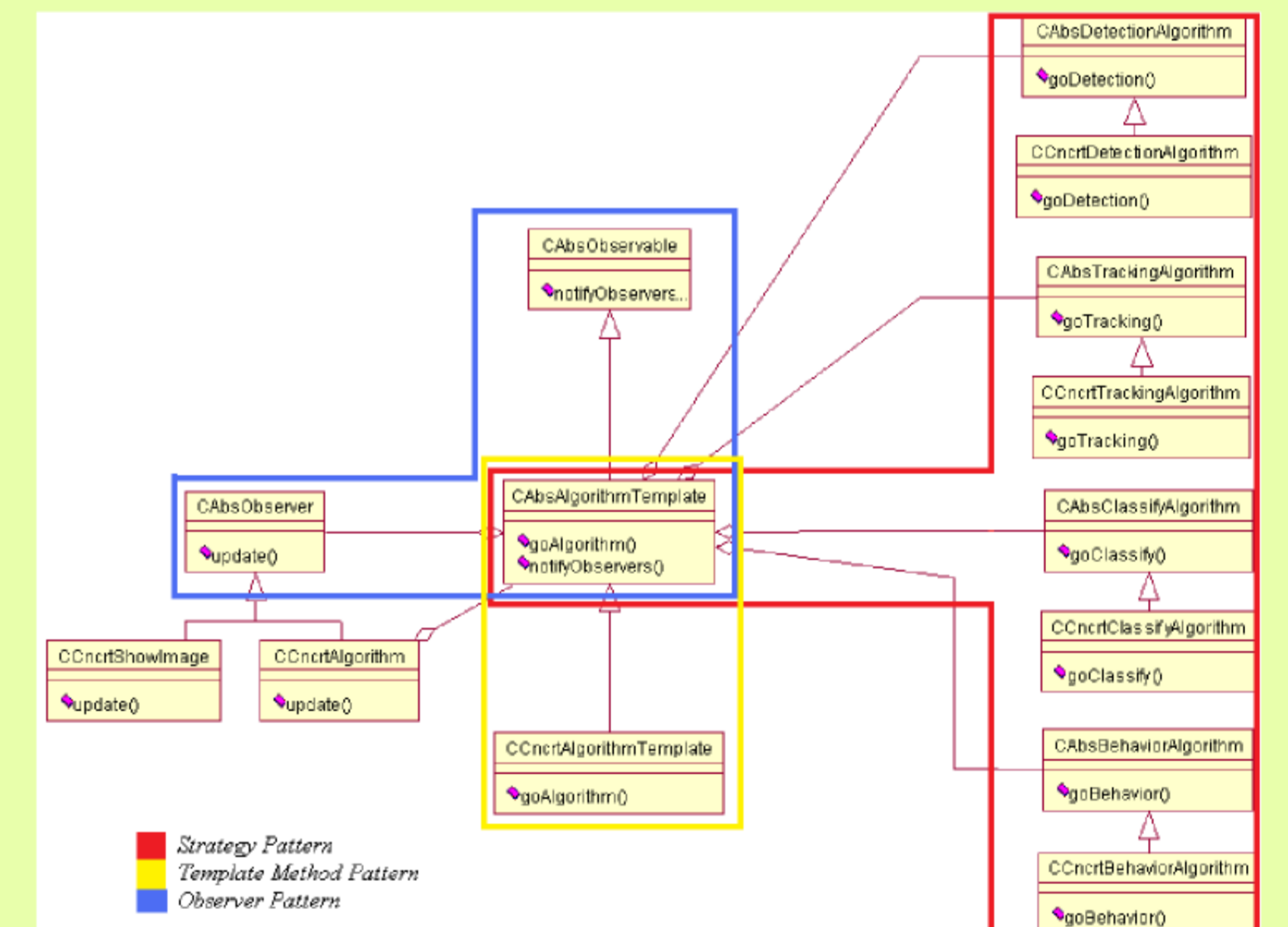


Figura 8. Sistema de procesamiento y salida de datos

## 7. Conclusiones

- Sistemas de vigilancia para medir el flujo de tráfico, obtener información sobre compradores en centros de atracciones, plazas comerciales.

- Aplicaciones militares como el patrullaje de bordes nacionales, proveer un perímetro seguro.

- La arquitectura funciona como un framework para el desarrollo de este tipo de aplicaciones.

## 8. Bibliografía

- [1] Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, Robert T. Collins, Alan J. Lipton and Lambert Wixson. A system for Video Surveillance and Monitoring, VSAM. Carnegie Mellon University, Pittsburgh PA and The Sarnoff Corporation, Princeton, NJ, 2000.
- [2] Marcos Zúñiga, Francois Brémont, Monique Thonnat. Video understanding framework for automatic behavior recognition, VSIP. 2004.
- [3] Ian Reid, Neil Robertson. A general method for human activity recognition in video. 2006.
- [4] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1995.
- [5] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. A pattern language: towns, buildings, construction. Oxford University Press, New York, 1977.
- [6] Alan Shalloway and James Trott. Design Patterns Explained: A New Perspective on Object-Oriented Design (2nd Edition) (Software Patterns Series). Addison-Wesley Professional, 2004.