

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División De Ciencias Básicas E Ingeniería

Ingeniería En Computación

Reporte Final De Proyecto Terminal

**Aplicación De Realidad Aumentada:  
Marcador De Señales De Tránsito**

Ayala Meza Fernando 205305172

10P

Septiembre 2010

Dr. Risto Fermin Rangel Kuoppa

## Introducción

El proyecto realizado pretende la creación de una aplicación que realice la simulación de un sistema de realidad aumentada la cual lleve a la identificación de señalamientos de tránsito en un flujo de video.

Actualmente existen técnicas las cuales ayudan o facilitan las actividades que realizamos cotidianamente, una de éstas es la visión por computadora o visión computacional, puede ayudar en las actividades donde hay cosas que pasan desapercibidas o no somos capaces de captar por descuido o falta de información.

Sabemos que la visión de los seres humanos está limitada a la banda visible del espectro electromagnético, mientras que las máquinas pueden percibir casi el espectro completo, así que les es posible procesar información la cual el ser humano no tiene conocimiento de ella.

Un campo adicional a la técnica de visión por computadora, es la realidad aumentada, la cual se refiere a un conjunto de dispositivos que añaden información virtual a la física ya existente (Figura 1).



Figura 1. Identificación de un edificio y su descripción a través de un sistema de realidad aumentada. “*How Stuff Works, 2001*”

Uno de los sectores que se pueden beneficiar por la realidad aumentada, es el del automóvil en todas sus áreas, desde la fabricación, hasta el apoyo al conductor de un vehículo. Esto es lo que se quiere lograr con el proyecto: demostrar que se puede asistir al conductor en la parte de la atención a las señales de tránsito que están en el camino, y así tratar de evitar accidentes o infracciones.

Lo que se quiere conseguir es crear una aplicación la cual detecte las señales de tránsito y dé aviso al conductor por medio de una proyección visual, para esto, se necesitan conocimientos acerca de procesamiento de imágenes, bases de datos, algoritmos de clasificación, además de matemáticas y computación en general. Este proyecto tiene la

posibilidad de ser expandido al procesamiento en tiempo real, así como agregar módulos para visión nocturna y detección de obstáculos, automóviles o peatones.

## Objetivo general

Con base en el objetivo general que fue: “Crear una aplicación de tipo realidad aumentada simulada para la detección de señales de tránsito en un flujo de video almacenado en un archivo”.

Se logró que el programa pudiera identificar los señalamientos de tránsito en base a la distribución de los puntos que forman la imagen del mismo, se utilizaron algoritmos de manipulación de imágenes para poder obtener un descriptor del señalamiento y por medio de comparación lograr identificarlo.

Este objetivo se obtuvo gracias a la elaboración de 6 módulos (diagrama 1) los cuales se encargan de una tarea específica, cada módulo requiere el cumplimiento previo del módulo anterior para el total cumplimiento de su función.

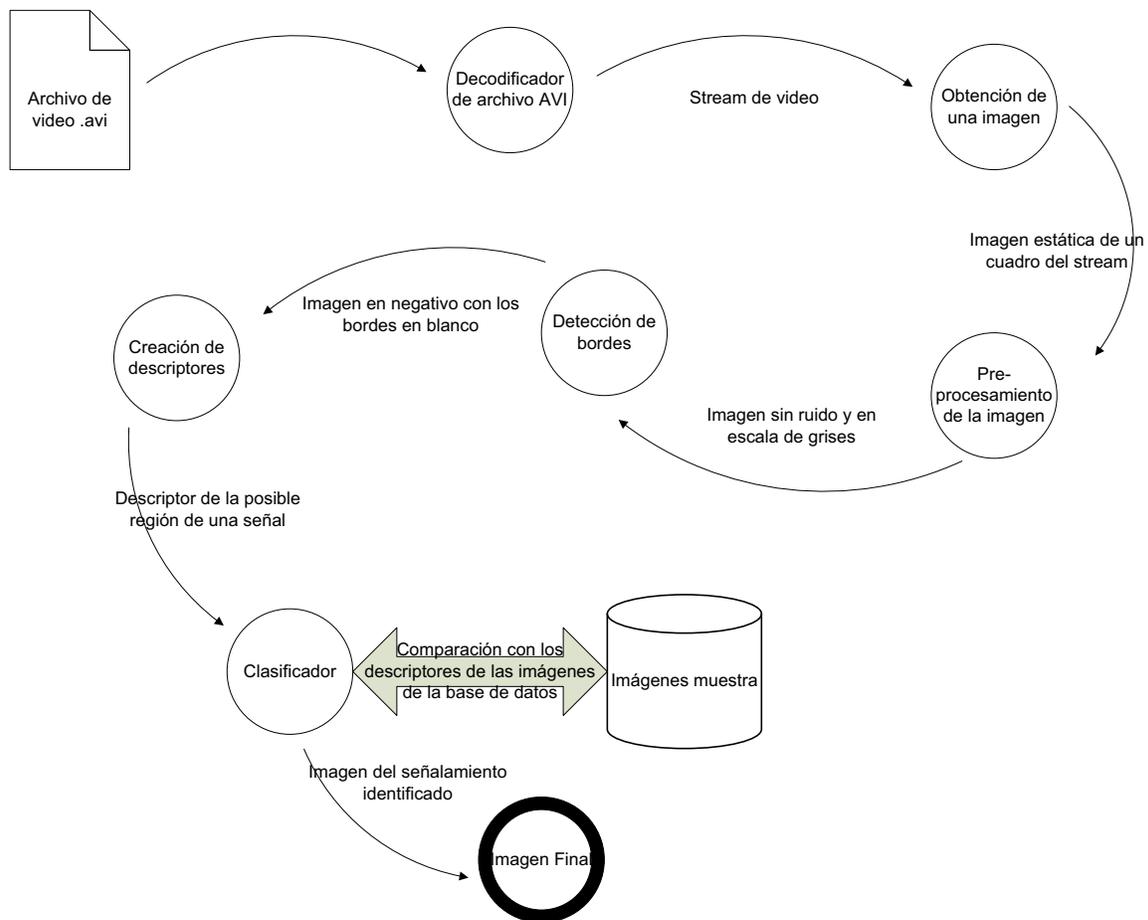


Diagrama 1. Módulos para la identificación de un señalamiento

**El módulo número 1**, se encarga de la lectura de un archivo de video (.avi) produciendo un flujo (stream).

**El módulo número 2**, se encarga de la utilización de un decodificador de AVI con lo cual se logra la captura de un cuadro del stream, con esta imagen se pasa al procesamiento de la misma.

**El módulo número 3**, se encarga de la limpieza de la imagen, quita el ruido de la misma y entrega una imagen más clara.

**El módulo número 4**, se encarga de generar una imagen la cual solo muestre los bordes de los elementos en la imagen ya sin ruido.

**El módulo número 5**, se encarga de la creación de un descriptor de cada una de las regiones de la imagen, de esta manera se puede hacer una comparación con los elementos de la base de datos para determinar si en ésta se encuentra un señalamiento.

**El módulo número 6**, se encarga de la clasificación de la región, determina a que señalamiento tiene un mayor parecido y regresa el señalamiento encontrado.

## **Objetivos particulares**

Involucrando cada uno de los objetivos particulares se obtuvo un resultado satisfactorio del proyecto. A continuación se describen cada uno de ellos, comentando el contenido o la falta de alguno de ellos.

### **Crear un módulo de lectura para la lectura de un archivo de video (.avi)**

La creación de este módulo se obtuvo con la ayuda de la biblioteca de DirectShow, la cual contiene herramientas para la lectura y reproducción de archivos de video, así, de esta manera se consiguió la lectura del archivo de video (.avi) y la generación de un stream, el cual es mostrado dentro de un PictureBox.

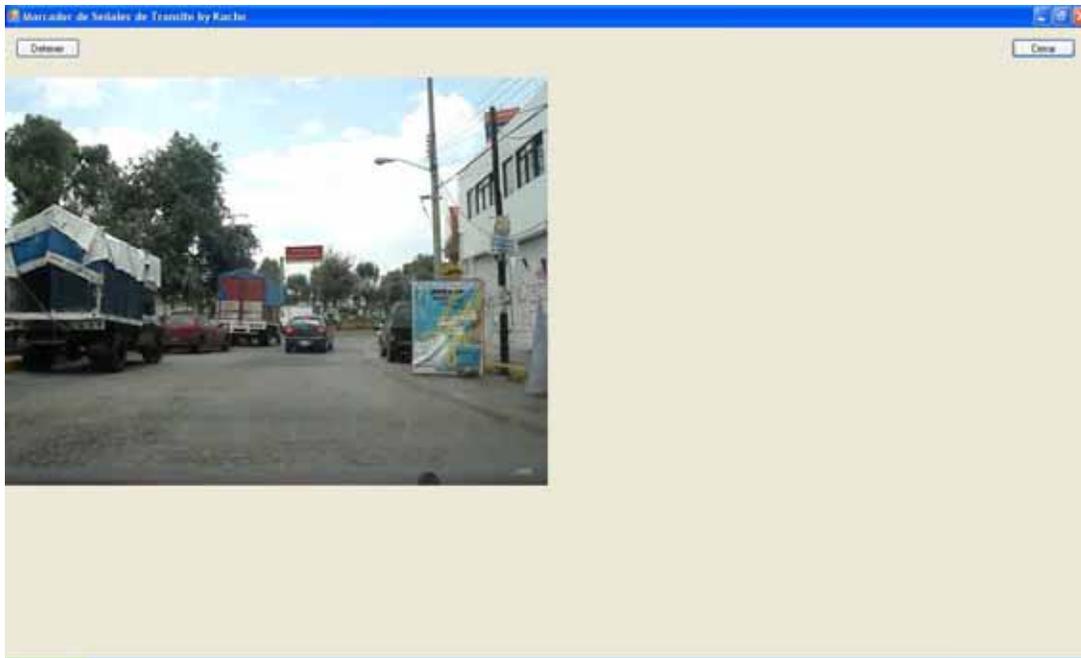


Figura 2. Reproducción de video

### **Utilizar un decodificador de AVI para obtener un cuadro del stream de video**

Con la ayuda de la misma biblioteca de DirectShow se logró la captura de un cuadro de video, se toma el stream de entrada que va a la PictureBox almacenando el buffer que llega a ella, consiguiendo así una instantánea del stream, la cual es almacenada como una imagen de tipo Bitmap.



Figura 3. Captura de cuadro

### **Crear un módulo de pre procesamiento para eliminar el ruido pimienta**

Para la creación de este módulo se utilizó una Difuminación Gaussiana (Gaussian Blur), la cual suaviza o desenfoca toda la imagen, eliminando el ruido y los detalles de la misma, dejando una imagen más propia para el módulo siguiente.



Figura 4. Imagen sin ruido

### **Crear un módulo de detección de bordes**

Este módulo comienza con la imagen obtenida del módulo anterior, primero se prepara convirtiéndola a una imagen en escala de grises, lo cual se consigue tomando cada pixel de la imagen y dividiendo cada componente de color en un factor.

Ya con esta imagen se pasó a utilizar el operador de Sobel el cual utiliza una convolución de matrices con la imagen para detectar los cambios de frecuencia en la misma, regresando una imagen con un fondo blanco y los bordes en negro.

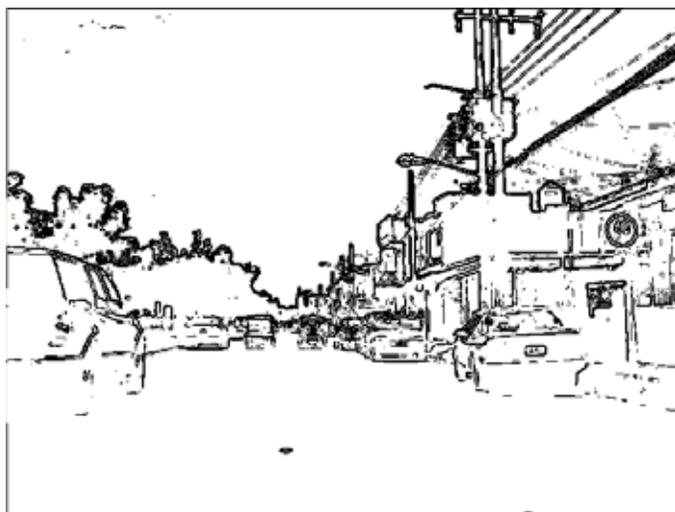


Figura 5. Imagen después del algoritmo de detección de bordes

### Crear un módulo de crecimiento de regiones

Este modulo no fue realizado, en su lugar se utilizó una serie de matrices de diferentes dimensiones para conseguir la localización de los señalamientos en la imagen.

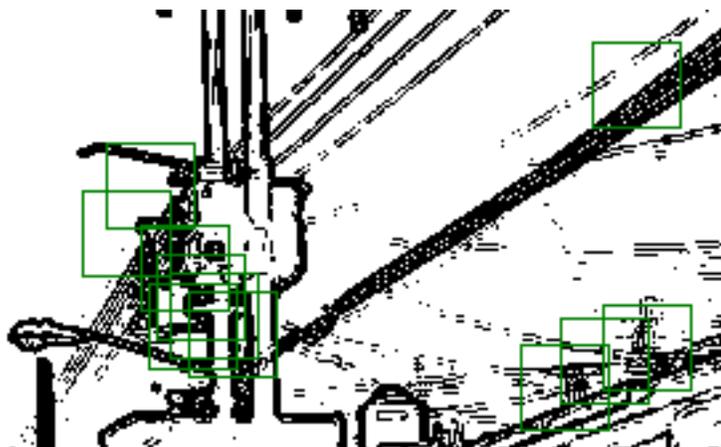


Figura 6. Marcado de las matrices con descriptores similares a los de las señales

### Crear un modulo clasificador

Con las matrices utilizadas se crearon descriptores de cada una de las regiones de la imagen, de esta manera se tuvo una manera de comparar cada una de ellas con cada uno de los descriptores de cada una de las imágenes de la base de datos, regresando el grado de pertenencia a cada una de ellas, teniendo así la región y la imagen base más parecidas, llegando a la conclusión de que ahí se encuentra ese señalamiento.



Figura 7. Detección y clasificación de la señal

## Conclusiones

La implementación de este trabajo resolvió el problema principal, que es poder detectar señales de tránsito en un archivo de video.

Se presentaron problemas por la lentitud del algoritmo, ya que revisa cada región de la imagen varias veces dependiendo de la dimensión de la matriz que se esté utilizando en ese momento, además del tiempo que se añade por la limpieza de la imagen y la detección de bordes de la misma, las cuales se hacen un pixel a la vez y revisando los pixeles que lo rodean.

Las pruebas del algoritmo se realizaron con fotografías y no con videos, con lo cual se eliminaba el tiempo desperdiciado en el procesamiento de imágenes que no contenían alguna señal en ella, sobre imágenes funcionó muy bien, en la mayoría de los casos detectó la señal correcta y en un tiempo moderado.

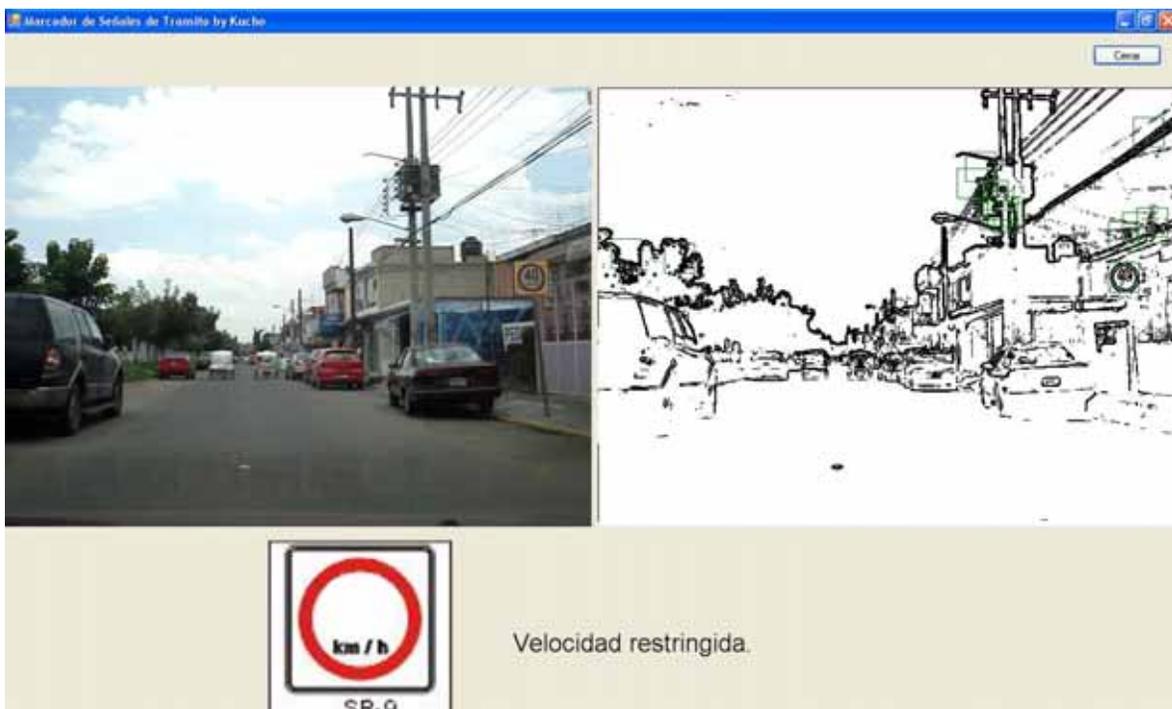


Figura 8. Detección de señal

En las ocasiones que tuvo problemas fue en los entornos con muchos distractores, aquéllos que en la imagen contenían mucha información o elementos, tales como: autos, anuncios, personas, etc. En estos casos se detectaron anuncios o algunos otros elementos como si fueran señales de tránsito dado que la distribución de los puntos en ellos regresaba datos muy parecidos a los calculados para las imágenes de las señales.

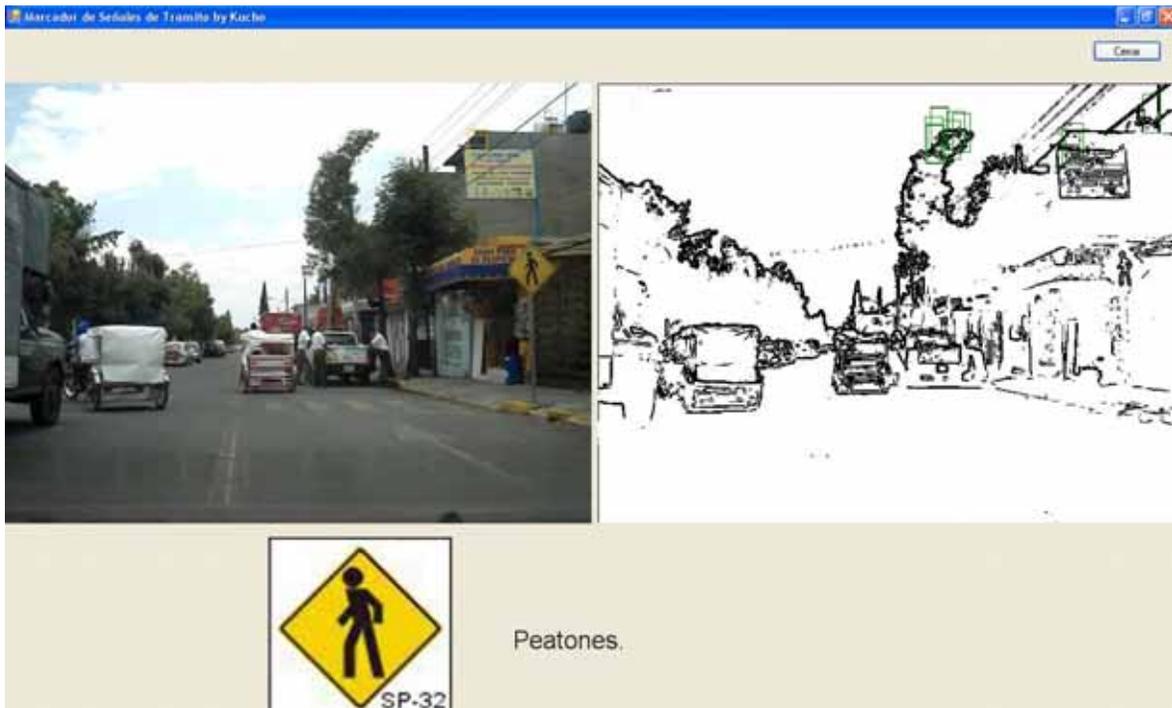


Figura 9. Error de detección en un entorno con mucha información en la imagen

También se tuvieron ciertos problemas con la falta de luz, ya que no se distinguía entre las señales y el fondo haciendo que se perdieran los bordes de la misma, por lo tanto no siempre se llegaba a la detección de esas señales.

En cuanto al procesamiento de la información incluida en la imagen, el mayor problema se detectó en condiciones climáticas de lluvia, los videos fueron tomados desde dentro del auto y el agua en el parabrisas creó un efecto el cual hacía que al momento de pasar el algoritmo de detección de bordes la imagen quedara prácticamente “limpia” (sin información), lo cual hizo imposible la detección de algún señalamiento.

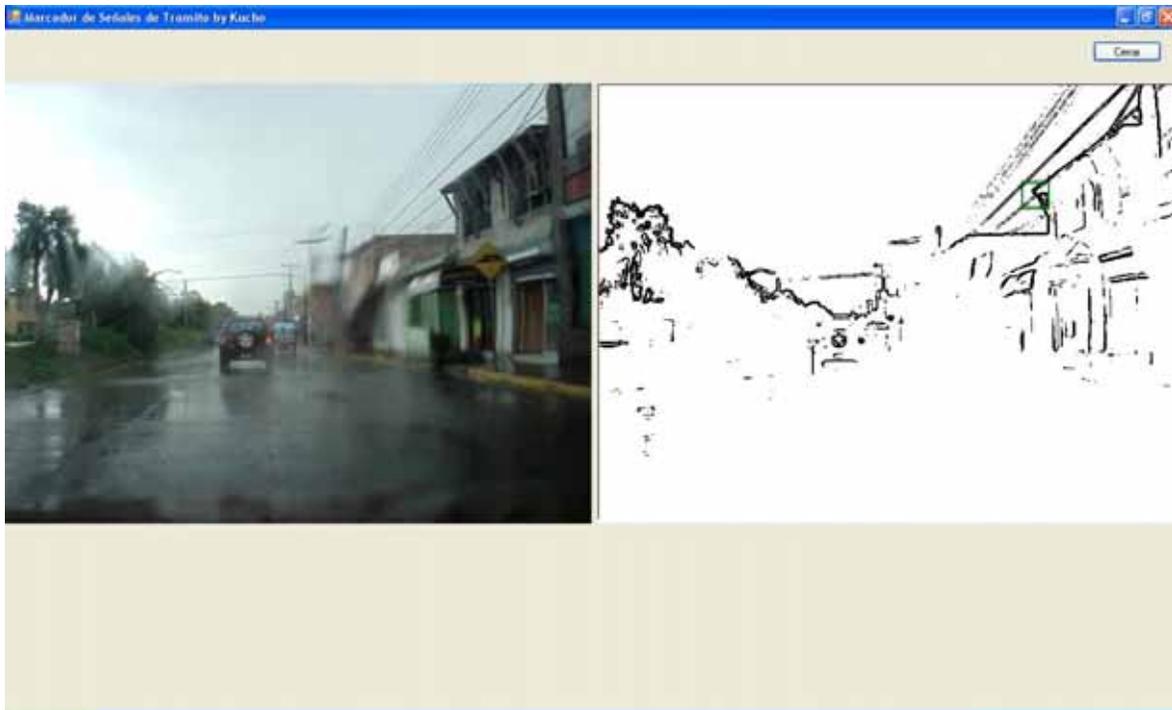


Figura 10. Falta de información en la imagen a causa de la lluvia

A una mayor cantidad de agua en el parabrisas, ocasionado por una lluvia más intensa, menos información queda en la imagen después de pasar el algoritmo de detección de bordes.

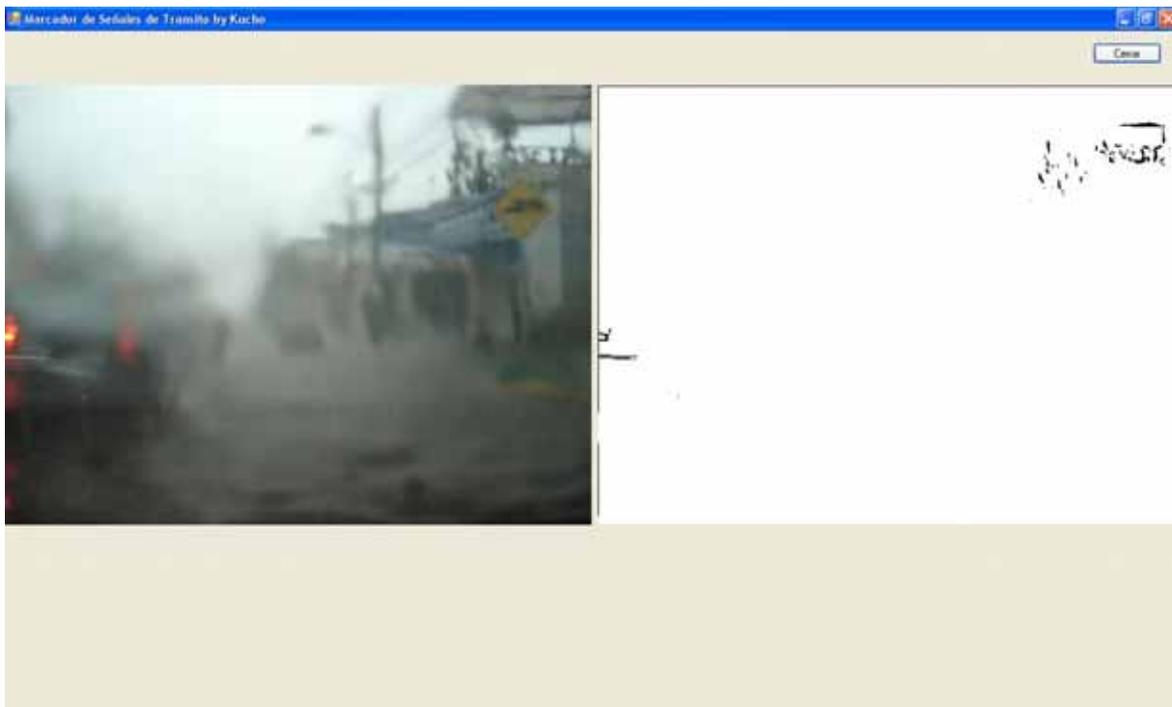


Figura 11. Imagen prácticamente sin información a causa de la lluvia

Otro problema se detectó al momento de ejecutar la aplicación ya completa ya que el hilo principal esperaba mucho tiempo por el que ejecuta la búsqueda del señalamiento, lo cual generaba una excepción en el hilo principal ya que por un largo periodo no recibía respuesta alguna del otro.

Para finalizar se puede decir que la aplicación funciona y es capaz de detectar señalamientos de tránsito en un flujo de video, con ciertos problemas como los descritos anteriormente, pero en un balance, la aplicación cumple con su función.