

Proyecto Final

Diaz Uboe, Pablo
Elli, Federico
Pomar, Federico

31 de agosto de 2016

Índice general

1. Introducción	3
2. Estado del arte	4
2.1. Plataformas similares	4
2.1.1. Servicio Meteorológico Nacional	4
2.1.2. Nullschool Earth	5
2.1.3. Windyty	6
2.2. Trabajos académicos	7
2.3. Conclusiones	8
3. Marco Teórico	9
3.1. DBSCAN	9
3.2. Flujo óptico	10
3.2.1. Ecuación del flujo óptico	11
3.2.2. Algoritmo de Lucas y Kanade	11
3.2.3. Problema de la apertura	12
3.2.4. Algoritmo de pareo de bloques	12
3.3. Canales infrarrojos	13
3.4. Conclusiones	14
4. Implementación	15
4.1. Algoritmo	15
4.1.1. Técnicas adicionales	15
4.1.2. Cálculo de trayectorias	16
4.1.3. Predicción	17
4.1.4. Lenguaje de programación y bibliotecas	17
4.2. Arquitectura	18
4.2.1. Base de datos	18
4.2.2. Obtención de imágenes	19
4.2.3. Servicios de cálculo: Movimiento y predicciones	20
4.2.4. API	20
4.2.5. Aplicación web	20
5. Resultados	23
5.1. Problemas	23
5.2. Casos de éxito	25
6. Conclusiones	27
6.1. Conclusiones generales	27
6.2. Posibles ampliaciones	27

<i>ÍNDICE GENERAL</i>	2
Bibliografía	28

Capítulo 1

Introducción

Desde la antigüedad, la meteorología ha fascinado al ser humano. El poder predecir el clima ha sido uno de los retos más grandes de la humanidad a través de la historia. El primer registro escrito data de alrededor del año 340 a.C. y se trata del libro “Meteorologica”, escrito por Aristóteles, en el que presenta observaciones y especulaciones acerca del origen de los fenómenos atmosféricos y celestes. Sin embargo, el uso de satélites para el estudio de los fenómenos climáticos es más reciente, ya que el primer satélite meteorológico fue lanzado en la década del sesenta.

Hoy en día, los modelos de predicción sobre imágenes satelitales son la base para muchos procesos de meteorología. Es indiscutible la importancia que éstas imágenes han tomado en los últimos años, desde la predicción del estado del clima diario hasta la de posibles catástrofes naturales, que permiten salvar miles de vidas en todo el mundo, son algunos de los usos que se le dan a dichas imágenes día a día.

En este proyecto se busca detectar de manera automática el movimiento de las nubes, a partir de imágenes satelitales, y con un modelo simple prever la posición y tamaño de las mismas para las siguientes horas.

Otro objetivo, es probar la eficacia de algoritmos que son utilizados en el campo del tratamiento de imágenes, especialmente para la detección de movimiento en vídeo, en el campo de la meteorología.

Por otro lado, es importante proveer una plataforma flexible sobre la cual puedan basarse otros proyectos para realizar nuevas ampliaciones en base a los datos provistos por este trabajo.

El objeto de estudio tratado en este proyecto está limitado a una sola región que comprende la mayoría del territorio argentino y chileno y una pequeña porción del territorio de Brasil. Una posible extensión del proyecto, propuesta más adelante, comprende la ampliación de esta región.

Capítulo 2

Estado del arte

En este capítulo se presentará el estado del arte, tanto teórico como práctico, en lo que respecta al seguimiento automático de nubes. Se estudiarán herramientas desarrolladas con un propósito similar e investigaciones relacionadas con la temática. Finalmente, se concluye con una breve reflexión sobre lo mencionado en este apartado.

2.1. Plataformas similares

Las plataformas presentadas a continuación constituyen parte del estado del arte en cuanto a la visualización del estado del clima. Para cada alternativa se describen sus características, fortalezas y debilidades. Además se incluye un breve análisis de su funcionalidad y de su interfaz.

2.1.1. Servicio Meteorológico Nacional

El servicio meteorológico nacional [9] posee una plataforma en la que se pueden visualizar imágenes satelitales infrarrojas captadas por el satélite geoestacionario GOES13. Dichas imágenes se encuentran espaciadas por un intervalo de aproximadamente una hora entre cada una. Como puede verse en la figura 2.1, la interfaz no se destaca por ser del todo amigable, aunque resulta bastante intuitiva. A la izquierda de la pantalla se pueden elegir distintas zonas y el tipo de imágenes disponibles. Sobre la imagen hay disponibles controles para seleccionar la cantidad de imágenes que se quieren ver y para pasar entre ellas.

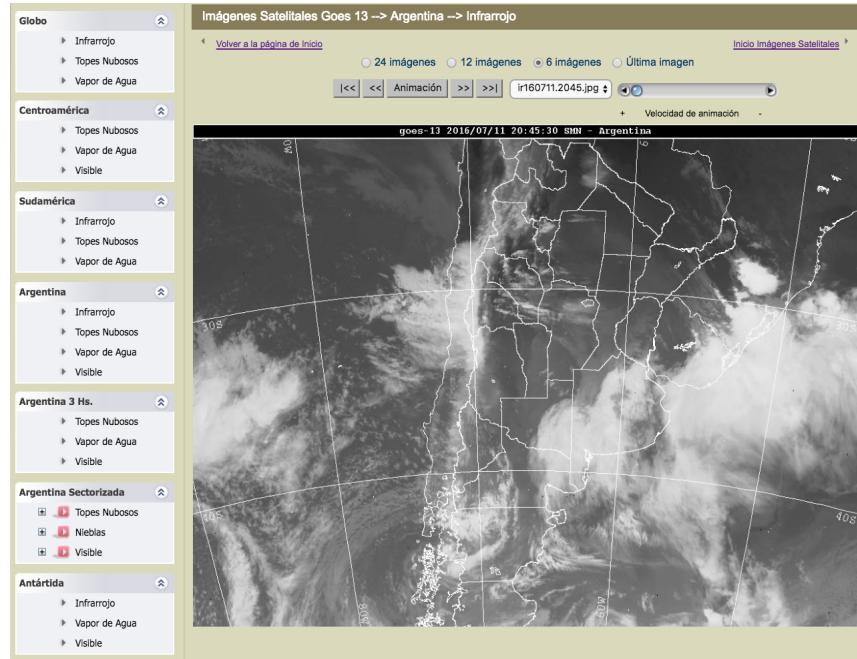


Figura 2.1: Captura de la página de visualización de imágenes infrarrojas del SMN [fecha de consulta: 7 de Julio de 2016]. Disponible en: <http://www.smn.gov.ar/vmsr/principal.php>

Cabe destacar que, aunque se hace foco en las imágenes de Argentina, existe la posibilidad de ver imágenes de todo el continente americano y la Antártida.

Además, el sistema permite visualizar hasta 24 anteriores e incluso se puede ver una pequeña animación de la secuencia de imágenes y configurar la velocidad con la que pasan.

Otro aspecto importante es que, además de las imágenes infrarrojas, se pueden visualizar imágenes de topes nubosos con sus respectivas temperaturas y de vapor de agua.

Como una cualidad negativa, se puede ver que no se proporciona ningún tipo de análisis sobre las imágenes infrarrojas ni ningún tipo de información complementaria.

Otra característica negativa de la interfaz es que cuando está seleccionada la opción “Última imagen”, los controles para pasar entre imágenes no tienen ninguna función lo que resulta bastante confuso, por lo que lo mejor quizás sería no mostrarlos hasta que se seleccione alguna opción en la que tengan sentido.

2.1.2. Nullschool Earth

Esta plataforma [26] brinda acceso a multitud de pronósticos, como por ejemplo corrientes oceánicas, temperatura del océano, oleaje, vientos e incluso información sobre partículas y químicos presentes en la atmósfera.

La interfaz de usuario resulta muy vistosa, aunque en principio no es demasiado intuitiva. En la figura 2.2 se ve que las opciones de configuración se encuentran un poco escondidas, teniendo que hacer click en el logo en la parte inferior izquierda para que aparezcan. Las mismas son muy completas ya que se puede cambiar desde la altitud de los vientos que se muestran hasta incluso distintas proyecciones del mapa si así se desea.

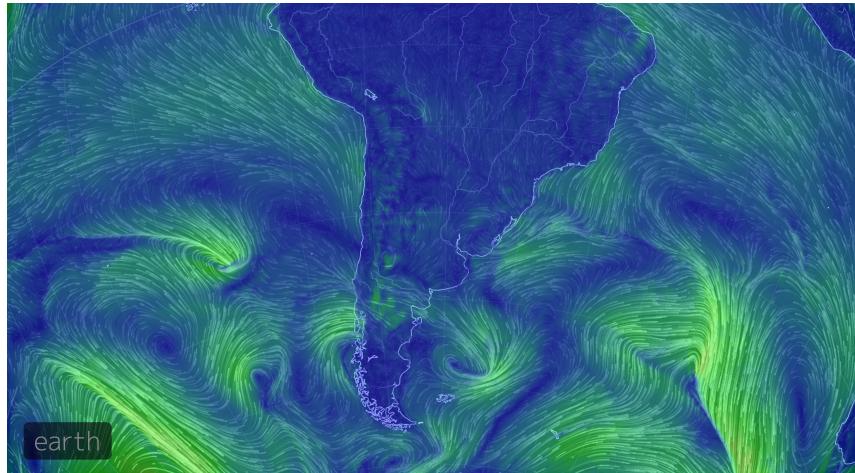


Figura 2.2: Captura del mapa de Nullschool Earth [26], sector Argentina. [fecha de consulta: 7 de Julio de 2016]. Disponible en: <https://earth.nullschool.net/>

El sistema resalta por la vasta cantidad de información que provee, es muy completa y diversa y se encuentra disponible para todo el planeta.

Por otro lado, es destacable la forma de visualizar la información, ya que las configuraciones son muy completas y satisfacen casi cualquier preferencia.

Como aspecto mejorable, se puede mencionar que el uso de colores no es demasiado claro. Las escalas de colores confunden y no brindan suficiente información. Las mismas no están numeradas lo que no ayuda a su fácil comprensión.

2.1.3. Windyty

Al igual que la anterior esta plataforma [28] brinda acceso a multitud de pronósticos distintos, como por ejemplo, viento, temperatura, nubosidad, lluvias, humedad, oleaje, temperatura del mar, nevadas y presión atmosférica.

La interfaz es muy vistosa y simple (Figura 2.3). Arriba a la izquierda muestra un pronóstico climático para los próximos 3 días de la ubicación actual, arriba a la derecha muestra una escala de colores, abajo a la derecha se observa el menú, el cual ofrece distintas opciones para visualizar la información y finalmente abajo a lo ancho de la pantalla aparece una línea de tiempo si se coloca el puntero encima, la

cual permite cambiar la fecha y hora. En el mapa también se ven pequeños triángulos que representan estaciones climáticas, las cuales si se les hace click se pueden ver las mediciones de las condiciones más recientes que han registrado.

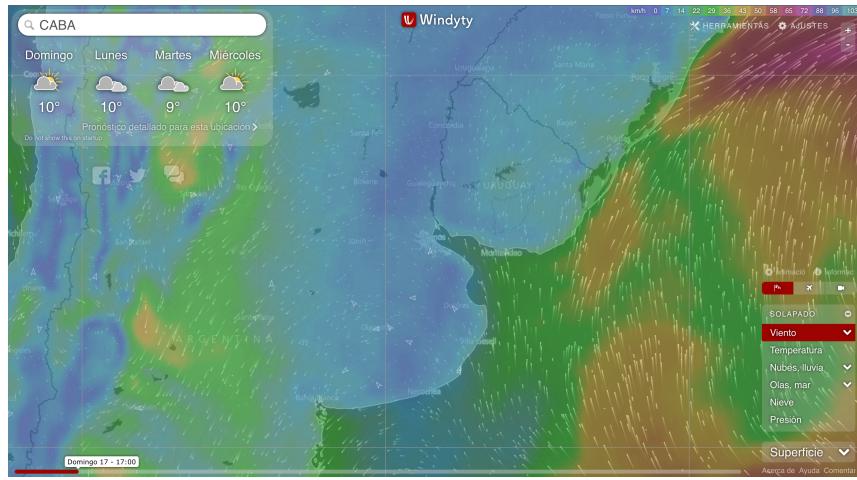


Figura 2.3: Captura del mapa de Windyty, sector Argentina. [fecha de consulta: 7 de Julio de 2016]. Disponible en: <https://www.windyty.com/-34.603,-58.382,6>

Entre lo más destacable del sistema, está la cantidad de opciones disponibles para visualizar la información y que la misma esté disponible para todo el planeta.

Por otro lado, es importante notar que se pueden elegir las unidades en las que se muestra la información y que además, es posible cambiar el tipo de mapa que hay en el fondo para que cuando se haga zoom se pueda ver con imágenes de satélite o bien un mapa con las calles de las ciudades.

Como aspecto mejorable, al igual que en la plataforma anterior, se puede mencionar que el uso de colores no es demasiado claro. Las escalas no usan colores apropiados para denotar una progresión de menor a mayor.

2.2. Trabajos académicos

El primer trabajo considerado fue “Extracting Cloud Motion from Satellite Image Sequences” [5]. En éste, los autores sugieren derivar el movimiento de las nubes a partir de un algoritmo de correlación. El proceso es el siguiente:

1. Se aplica un umbral automático de múltiples niveles para separar las características de las nubes del fondo.
2. Se correlacionan ambas imágenes utilizando un algoritmo de pareo de bloques [1]. El mismo genera vectores de movimiento.
3. Se mitiga el ruido vectorial por el pareo incorrecto utilizando un filtro de la mediana.

El trabajo es uno de una serie de investigaciones relacionadas al análisis espaciotemporal de imágenes satelitales.

Los buenos resultados obtenidos por los autores indican que es una opción factible y motivaron la utilización de algoritmos similares en este proyecto.

Otro trabajo que resulta de interés es “The Feasibility of an Optical Flow Algorithm for Estimating Atmospheric Motion” [6]. Éste analiza la factibilidad de medir movimientos atmosféricos utilizando un algoritmo de flujo óptico tradicional, el de Lucas y Kanade [24], en contraposición con algoritmos de correlación más tradicionales como el que finalmente fue utilizado.

Los autores concluyen en que este método es una alternativa factible a los algoritmos de correlación, los resultados presentados son buenos si y solo si se dispone de imágenes satelitales cuya frecuencia temporal es alta (entre 5 y 10 minutos entre cada una), sin embargo no es una alternativa viable en

este caso puesto que en la región geográfica estudiada en este proyecto dicha frecuencia no se encuentra disponible.

2.3. Conclusiones

El servicio meteorológico nacional [9] no dispone de herramientas para la visualización del movimiento de las nubes. En contraposición, herramientas como Windyty [28] y Nullschool Earth [26] parecen estar desarrolladas con fines más comerciales que científicos. Una de las metas del presente trabajo es combinar lo mejor de ambos mundos. Por otro lado, los resultados positivos de las pruebas en el campo de la meteorología de algoritmos de detección de movimiento utilizados en tratamiento de imágenes motivaron a que este fuera el camino a seguir en el desarrollo de este proyecto.

Capítulo 3

Marco Teórico

En este capítulo se introducirán los temas y conceptos necesarios para la correcta comprensión del tema y apreciación de las dificultades contenidas en la resolución del problema. Se comienza describiendo el algoritmo DBSCAN [11] ya que fue una de las primeras alternativas analizadas. Se continuará explicando algunos conceptos necesarios para entrar en contexto y entender más profundamente el funcionamiento del algoritmo implementado, como el concepto de flujo óptico, los problemas y algoritmos asociados a éste. Finalmente se estudiarán los datos de entrada, las imágenes satelitales producidas por los distintos sensores enfocados hacia la atmósfera terrestre.

3.1. DBSCAN

Density-based spatial clustering of applications with noise [11] es un algoritmo de clustering de datos basado en la densidad.

Dado un conjunto de puntos en un espacio, éstos son clasificados en puntos de núcleo, alcanzables y outliers de la siguiente manera:

- Un punto p es de núcleo si al menos k puntos se encuentran a una distancia menor a ε de él (incluyéndolo). Éstos se denominan *directamente alcanzables* desde p . Por definición, ningún punto es *directamente alcanzable* desde uno que no es núcleo.
- Un punto q es alcanzable desde p si existe un camino p_1, \dots, p_n con $p_1 = p$ y $p_n = q$, donde cada p_{i+1} es directamente alcanzable desde p_i (todos los puntos en el camino deben ser de núcleo a excepción de q que puede no serlo).
- Todos los puntos no alcanzables desde algún otro son outliers.

Entonces, si p es un punto de núcleo, forma un cluster junto con todos los alcanzables desde él. Cada cluster tiene al menos un punto de núcleo; puede tener algunos que no lo sean, pero forman parte del borde ya que no pueden ser usados para alcanzar a otros.

DBSCAN requiere dos parámetros: ε es la distancia máxima a la que un punto puede ser considerado vecino de otro y k es la cantidad mínima de puntos vecinos requeridos para formar una región densa. El algoritmo empieza con un punto arbitrario que no haya sido visitado. Se busca su vecindad y si contiene suficientes vecinos se crea un cluster, sino el punto se marca como ruido y se continúa con el próximo no visitado. Sin embargo, éste puede llegar a pertenecer a una vecindad de otro punto que si resulte suficientemente grande y por lo tanto ser parte de un cluster.

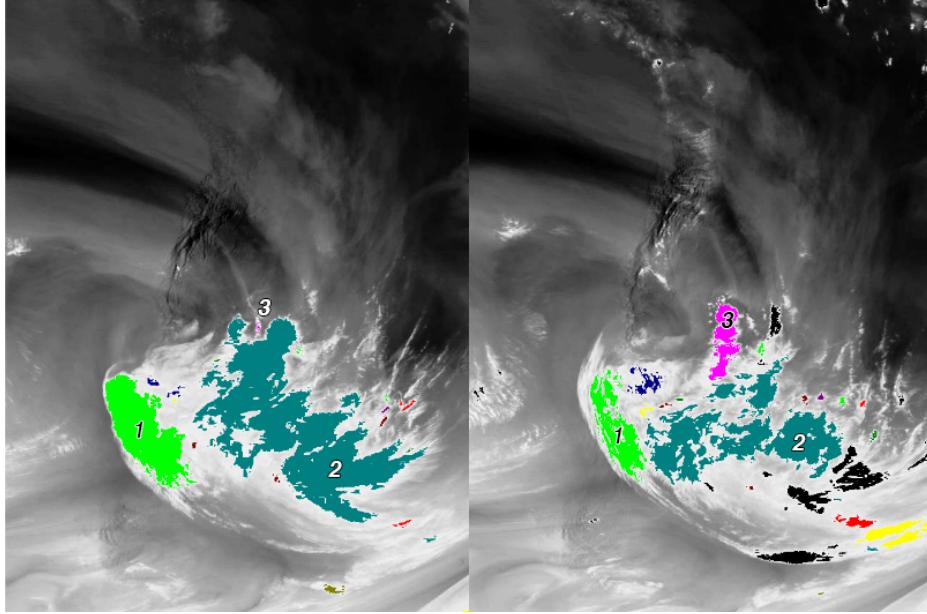


Figura 3.1: Secuencia de imágenes procesadas con DBSCAN.

El resultado que se pretendió lograr utilizando este algoritmo fue que cada nube formara un cluster, el cual se podría seguir a través de la secuencia de imágenes para obtener una estimación del movimiento. Sin embargo, esto presentó algunos problemas. En primer lugar, la volatilidad de los clusters entre imágenes subsecuentes hace que extraer información sobre movimiento resulte difícil. En segundo lugar, hay ocasiones en las que parte de un cluster se convierte en otro, lo que imposibilita que los resultados se correspondan con el movimiento real del sistema. Esto se puede observar en la Figura 3.1, donde parte del cluster 1 (en verde claro) se convierte en el 2 (en verde oscuro), y parte del 2 se convierte en el 3 (en rosa). Por último, hay numerosos clusters que aparecen en la segunda imagen que no pueden parearse con ninguno de la anterior y, por ende, no hay información sobre su movimiento. Estos últimos se observan en la Figura 3.1 en color negro. Estos problemas, sumado a que se lograron mejores resultados con otras alternativas, llevaron a que este algoritmo fuera finalmente descartado.

3.2. Flujo óptico

El flujo óptico es el patrón de movimiento de los objetos a través del campo visual del observador. Se introduce por primera vez en el campo de la psicología para describir la percepción del movimiento en la corteza visual [16]. Puede describirse matemáticamente como un campo vectorial sobre un espacio de dos dimensiones (Figura 3.2).

Existen dos técnicas fundamentales para la estimación del flujo óptico a partir de una secuencia de imágenes: la regresión del vector velocidad a partir de la ecuación del flujo óptico, utilizando, por ejemplo, el algoritmo de Lucas y Kanade [24]; así como la correlación de características [6]. Para el propósito de este trabajo se utilizará el campo vectorial del para estimar la rapidez y dirección de las masas de aire bajo la suposición de que el movimiento de las características nubosas es representativo de estos parámetros.

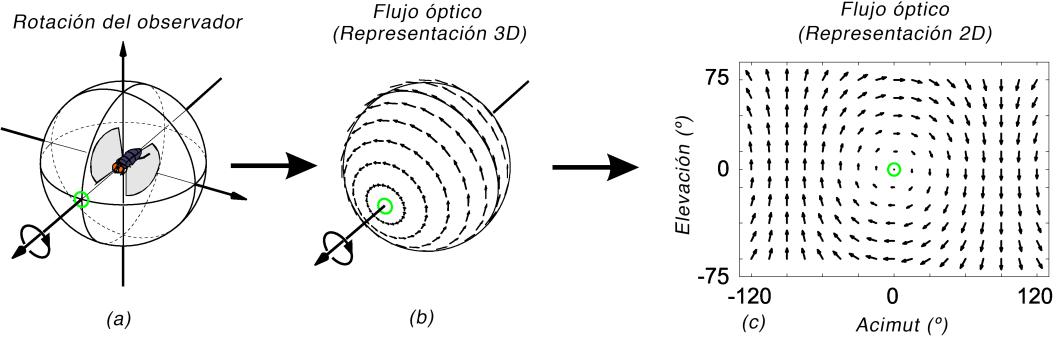


Figura 3.2: Descripción gráfica del flujo óptico: (a) Un sujeto posicionado en el interior de una esfera; (b) La esfera rota sobre su eje de profundidad, desde el punto de vista del sujeto; (c) Campo vectorial que indica el desplazamiento en un diferencial de tiempo de cada uno de los puntos de la esfera, proyectados en el campo visual del observador. Imagen tomada de [19].

3.2.1. Ecuación del flujo óptico

La ecuación del flujo óptico o ecuación de constancia de brillo [3] se basa en la hipótesis de la conservación de la intensidad luminosa emitida o reflejada por un objeto en movimiento:

$$I(x, y, t) \approx I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (3.1)$$

donde $I(x, y, t)$ es la intensidad luminosa en cada punto de la imagen a lo largo del espacio y el tiempo, puesto que se está hablando de movimiento, y los incrementos se denotan con Δ . Se introduce esta ecuación puesto que es condición necesaria para el método de Lucas y Kanade [24], y también lo será para el funcionamiento de la técnica que se utilizará en este trabajo. Sin embargo, se encontrarán condiciones meteorológicas que lo violan, por ejemplo, precipitaciones rápidas o formaciones de nubes repentinas. Estos detalles se analizarán más adelante.

3.2.2. Algoritmo de Lucas y Kanade

Uno de los algoritmos más mencionados en la bibliografía para la determinación del flujo óptico es el de Lucas y Kanade [24]. A partir de la ecuación (3.1) presentada en la sección 3.2.1 se puede derivar la siguiente expresión:

$$\vec{v} \cdot \nabla I^T = -\frac{\partial I}{\partial t}$$

donde $I(x, y, t)$ es la intensidad de la imagen en función del tiempo, $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$ es el gradiente de I y \vec{v} es la velocidad de desplazamiento de un punto de la imagen. Discretizando la ecuación se tiene:

$$\vec{v} \cdot (I_x, I_y)^T = -I_t$$

donde I_x , I_y e I_t , son diferencias discretas de la imagen y T es la traspuesta. Si adicionalmente se quiere calcular \vec{v} a considerando datos del entorno del punto, se puede hacer mediante la resolución del sistema:

$$\begin{bmatrix} I_x(q_1) & I_y(q_2) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

donde q_1, q_2, \dots, q_n con $q_i = (x_i, y_i)$ pertenecen al entorno del punto donde se desea calcular el vector \vec{v} .

Finalmente, puede resolverse el sistema sobredeterminado con algún método como el de mínimos cuadrados.

Solamente del enunciado del algoritmo se desprenden dos problemas. Debido a la localidad del sistema de ecuaciones en diferencias, la frecuencia temporal de las imágenes debe ser elevada, es decir, el intervalo de tiempo entre cada una debe ser reducido. Es porque no se dispone de esta frecuencia de imágenes satelitales que este método fue descartado. Adicionalmente, puede verse que solamente el entorno del punto es utilizado como dato para estimar la dirección de movimiento lo que da origen al problema de la apertura, el cual se tratará en la siguiente sección.

3.2.3. Problema de la apertura

Al poseer únicamente información sobre un punto y su entorno, es imposible inferir el movimiento global del objeto. El problema es universalmente conocido y puede ejemplificarse de manera muy simple: tan sólo basta colocar un cartón con un pequeño agujero sobre una diagonal, desplazarlo perpendicularmente a la misma e intentar percibir la dirección de movimiento de la apertura (Figura 3.3). Se notará que hacerlo sin referencias del entorno no es posible.

De la misma manera, un algoritmo que solamente considera información local, píxel a píxel, o bloque a bloque, no puede ofrecer como resultado información sobre el desplazamiento global del objeto que se está analizando.

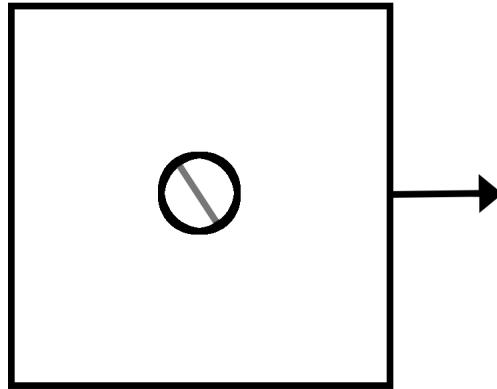


Figura 3.3: Problema de la apertura. Únicamente a partir del movimiento de la diagonal a través del agujero realizado en el cartón es imposible determinar el movimiento del cartón respecto al fondo.

Lucas y Kanade [24] sugieren aumentar la cantidad de píxeles utilizada en la regresión lineal. Para el caso del algoritmo de pareo de bloques, que se presentará en la siguiente sección, equivale a agrandar el tamaño del bloque. De todas maneras, las nubes conforman un cuerpo no rígido, que puede sufrir expansiones o contracciones. En estos casos resulta interesante analizar el movimiento local, puesto que, en la realidad los procesos meteorológicos no son homogéneos y están plagados de variaciones locales.

3.2.4. Algoritmo de pareo de bloques

A continuación se explica la variante del algoritmo de pareo de bloques [1] utilizado en la implementación de este trabajo. Dado un conjunto de píxeles $B_1 = \{(x, y) / x \in [x_1 - \Delta x, x_1 + \Delta x] \wedge y \in [y_1 - \Delta y, y_1 + \Delta y]\}$, de centro $p_1 = (x_1, y_1)$, que se denominará bloque B_1 , se busca encontrar B_2 , dentro del siguiente cuadro, con centro en $p_2 = (x_2, y_2)$ tal que $E(B_1, B_2)$ sea mínima. Siendo $E(B_1, B_2)$ una medida del

error entre los bloques. Para garantizar la ocurrencia de un mínimo, se consideran posibilidades únicamente dentro de una ventana de búsqueda, de tamaño parametrizable, que representa el entorno del punto. (Figura 3.4)

Definimos entonces:

$$\hat{M}(x_1, y_1) = (x_2 - x_1, y_2 - y_1)$$

El estimador del desplazamiento, dato del cual se puede derivar, por ejemplo, la velocidad de desplazamiento en dicho punto, $\vec{v} = \frac{\hat{M}}{\Delta t}$.

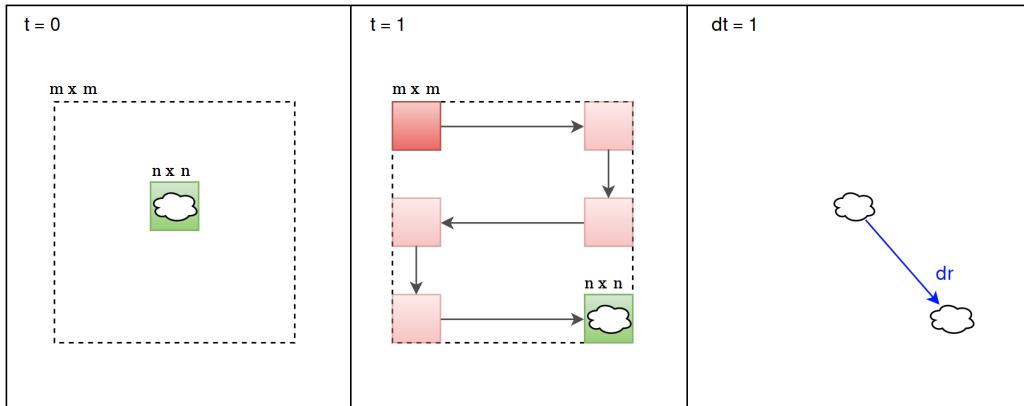


Figura 3.4: 1) B_1 centrado en una característica de la imagen 1. 2) Se comparan bloques B_2 dentro de la ventana hasta encontrar el de mínimo error, dentro de la imagen 2. 3) Queda determinado el vector $M(x_1, y_1)$ de movimiento.

La medida del error $E(x, y)$ utilizada fue el error cuadrático de la diferencia píxel a píxel dentro del bloque:

$$E(B_1, B_2) = \sum_{i,j} (I_1(x_1 + i, y_1 + j) - I_2(x_2 + i, y_2 + j))^2, \quad i \in [-\Delta x, \Delta x], \quad j \in [-\Delta y, \Delta y]$$

3.3. Canales infrarrojos

El satélite sensa energía electromagnética de cinco diferentes longitudes de onda. Las más utilizadas en pronósticos del clima son la infrarroja y la visible.

En el caso de los canales infrarrojo, el satélite sensa energía como calor. La superficie de la Tierra, junto con las nubes y la atmósfera absorben alrededor de la mitad de la energía solar entrante. Parte de esta energía absorbida es luego reemitida como calor. Este calor es captado por los diferentes canales infrarrojos del satélite.

Los canales utilizados en este trabajo son los siguientes:

IR2 (3,9μm): Infrarrojo de longitud de onda corta, la mayoría de la energía sensada es emitida por la superficie de la Tierra y las nubes debido a su temperatura. Durante el día ocurren reflejos de una parte de la radiación solar causando efectos no deseados en la imagen.

IR3 (6,7μm): En este canal, la energía emitida por la superficie de la tierra es prácticamente nula. Casi la totalidad es emitida por las capas medias de la atmósfera, con excepción de las porciones nubladas de la imagen donde la energía proviene de los topes de las nubes. Este canal también es llamado de vapor de agua ya que éste es el principal constituyente de las capas de las que proviene la mayoría de la radiación en esta longitud de onda, por lo tanto la temperatura de las

mismas (y por consiguiente la cantidad radiación que emiten) depende de la cantidad de vapor de agua presente.

IR4 ($10,5\mu m$): Infrarrojo de longitud de onda larga, la mayoría de la energía es emitida por la superficie de la tierra o bien los topes nubosos. La emisión de radiación por parte de la atmósfera en este espectro es mínima.

Una de las principales ventajas de los canales infrarrojos por sobre el visible es que pueden captar energía durante la noche, por lo que estas imágenes están disponibles durante las 24 horas del día.

3.4. Conclusiones

La detección automática de movimiento de nubes en imágenes satelitales resulta un problema complejo. Además, la meteorología es una ciencia que abarca fenómenos de gran escala, relacionados entre si alrededor del mundo y con muchísimas variables, lo que dificulta en gran medida su correcta comprensión. En este apartado se trató de explicar algunos conceptos necesarios para la correcta apreciación de las decisiones de implementación tomadas, las cuales se detallan en el próximo capítulo.

Capítulo 4

Implementación

En este capítulo se abordarán distintos aspectos de implementación de la plataforma.

Se analizarán primero detalles técnicos sobre el algoritmo, mejoras realizadas sobre el mismo y los datos más elaborados, como las trayectorias y la predicción, que serán derivadas del mismo.

En segundo lugar se hablará sobre los detalles de arquitectura, pasando por las diversas componentes que la constituyen.

4.1. Algoritmo

El algoritmo desarrollado es una variante del algoritmo de pareo de bloques, descripto en la sección 3.1.4 sobre el cual se aplicaron variaciones para mejorar los datos de entrada. Para las mismas se aprovechó la disponibilidad de imágenes de distintos sensores obtenidas a lo largo de los meses de Junio y Julio de 2016 a razón de una imagen por hora, en promedio.

4.1.1. Técnicas adicionales

Por sobre la base del algoritmo pueden realizarse una serie de mejoras que permiten que los resultados del mismo estén mejor ajustados al movimiento real de las nubes:

Detección de bordes: Es posible realizar una detección de bordes previa al procesamiento. Esto presenta una alternativa al cálculo convencional, permitiendo correlacionar las características a partir de la información de los bordes y no de la intensidad de la imagen. Está inspirado en la idea de que la intensidad de una imagen no está relacionada al detalle contenido en la misma, como en los casos en los que un área grande es uniforme. Para detectar un borde se calcula una aproximación discreta la norma del gradiente $\|\nabla I\|$ de la imagen, utilizando el siguiente procedimiento:

$$\|\nabla I\| = \|(I_x, I_y)\| = \sqrt{I_x^2 + I_y^2}$$

donde I_x e I_y son las derivadas en x y en y de la imagen calculadas mediante la convolución:

$$I_x = [-1 \ 0 \ 1] * I$$

$$I_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * I$$

Remoción de fondo: A partir de la recolección de una cantidad grande de imágenes puede extraerse el fondo de la escena. Esta nueva imagen, que es calculada utilizando una herramienta provista,

sirve para independizar del cálculo aquellas características que están siempre presentes y corresponden a la geografía subyacente y no a las formaciones nubosas del momento. La extracción consiste en tomar el valor mínimo de intensidad de cada píxel:

$$I_F(x, y) = \min \{I_t(x, y)/t \in \Gamma\}$$

Siendo I_F el mapa de intensidad de la imagen de fondo final, I_t el mapa para un tiempo t y Γ el conjunto de tiempos de las imágenes a analizar. Para realizar los experimentos se utilizó un fondo creado a partir de 2900 imágenes para cada sensor.

Combinación de resultados: Una vez calculados los movimientos a partir de cada uno de los sensores, se realiza un promedio ponderado de los sensores en función de su fiabilidad. Por ejemplo, para combinar sensores IR2, IR3 e IR4:

$$\begin{aligned} \hat{M}(x, y) &= \hat{M}_{IR2}(x, y) * w_{IR2} + \hat{M}_{IR3}(x, y) * w_{IR3} + \hat{M}_{IR4}(x, y) * w_{IR4} \\ w_{IR2} + w_{IR3} + w_{IR4} &= 1 \end{aligned} \quad (4.1)$$

donde $0 \leq w_{IR} \leq 1$ es el peso que se le desea dar al sensor IR y $\hat{M}_{IR}(x, y)$ es el vector resultado de correlacionar imágenes consecutivas del sensor IR . El valor de los pesos se determinó de forma empírica, utilizando como referencia la información cada uno de los canales infrarrojos introducida en la sección 3.3. En particular para realizar la estimación exhibida se utilizaron los pesos $w_{IR2} = 0,1$, $w_{IR3} = 0,5$ y $w_{IR4} = 0,4$; IR2 tiene un peso considerablemente menor en la combinación porque es afectado en mayor medida por la reflexión de parte de la radiación solar en la superficie de la Tierra durante el día. Este problema, al que llamamos variación día-noche, se discutirá en la sección 5.1.

4.1.2. Cálculo de trayectorias

Una vez calculados los desplazamientos por cada par de estados meteorológicos consecutivos, es posible, dado un punto $p_0 = (x_0, y_0)$, estimar la trayectoria $T = \langle p_0, p_1, \dots, p_n \rangle$ de una partícula virtual que viaja a través del área en estudio. Puesto que en principio sólo se dispone de p_0 , deben estimarse las posiciones de los puntos siguientes utilizando los movimientos $\hat{M}_{j,k}$ calculados por el algoritmo de pareo de bloques [1] explicado en el capítulo anterior. A partir de esta información se estiman los puntos de la siguiente manera:

$$\hat{p}_k = \begin{cases} p_0 & k = 0 \\ p_{k-1} + \hat{M}_{k-1,k}(p_{k-1}) & k \neq 0 \end{cases}$$

Puesto que la información hasta este punto es discreta, para poder modelar el movimiento de la partícula en un continuo y además poder realizar una extrapolación para la predicción - que se verá en la sección 4.1.3 - resulta conveniente utilizar una aproximación polinómica de grado m para poder obtener así una función continua de la posición en el tiempo.

Se define:

$$\begin{aligned} p(t) &= \langle x(t), y(t) \rangle \\ x(t) &= a_0 + a_1 t + \dots + a_m t^m \\ y(t) &= b_0 + b_1 t + \dots + b_m t^m \end{aligned} \quad (4.2)$$

y se calculan los coeficientes de los polinomios utilizando la técnicas de mínimos cuadrados. Se resuelve el sistema de ecuaciones:

$$\left[\begin{array}{cccc} 1 & t_0 & \cdots & t_0^m \\ 1 & t_1 & \cdots & t_1^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & \cdots & t_n^m \end{array} \right] \left[\begin{array}{cc} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_m & b_m \end{array} \right] = \left[\begin{array}{cc} x_0 & y_0 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{array} \right]$$

Los resultados son finalmente dibujados con una biblioteca gráfica para luego poder ser exhibidos en la aplicación web.

4.1.3. Predicción

Como ya se mencionó en la sección 4.1.2, las curvas de desplazamiento de las partículas a lo largo del tiempo pueden ser utilizada para realizar una predicción mediante una extrapolación de las mismas. Además, se puede utilizar la imagen del último estado meteorológico para construir un posible escenario futuro, desplazando bloques a partir de ellas.

El proceso de cálculo es el siguiente:

- Se escoge:
 - Una imagen en un tiempo t_p a predecir
 - Un intervalo de tiempo $[t_i, t_f]$ que determinará un conjunto de imágenes en tiempos $t_i \leq t_0 < t_1 < \dots < t_n \leq t_f < t_p$.
 - Un conjunto de puntos $\{p_1, \dots, p_k\}$ distribuidos homogéneamente sobre la zona abarcada por el sensor.
- Se calculan las trayectorias del conjunto de puntos, tal como fue descripto en la sección 4.1.2. Obteniéndose así un conjunto de pares de polinomios como los de la expresión 4.2.
- Se evalúan dichos polinomios en t_p , obteniéndose así los puntos finales en el tiempo a predecir.
- Se desplaza un área de $m \times m$ píxeles con centro $p(t_f)$ hacia $p(t_p)$ por cada trayectoria.
- Se promedia cada área con cada una de las que se le superpongan para así lograr la luminosidad final de cada zona a predecir.

Para tener una mejor apreciación de la precisión de la predicción, se decidió hacer disponible también una imagen que corresponde a la diferencia entre la predicción y la imagen real obtenida del satélite. La misma se calcula de la siguiente manera:

$$I_D^*(x, y) = |I_P(x, y) - I_R(x, y)|$$

$$I_D(x, y) = \frac{I_D^*(x, y) - \min\{I_D^*(x, y)\} / (x, y) \in Dom_{I_D^*}\}}{\max\{I_D^*(x, y)\} / (x, y) \in Dom_{I_D^*}\}}$$

donde I_D es la intensidad final de la diferencia normalizada, I_D^* lo es sin normalizar, I_P es la intensidad de la predicción, I_R es la intensidad de la imagen real y Dom es el dominio de la función.

Para los experimentos se tomó un tiempo de extrapolación $t_p - t_f$ de dos horas, y un intervalo de tiempo $t_f - t_i$ de seis.

4.1.4. Lenguaje de programación y bibliotecas

La implementación fue realizada en el lenguaje de programación Python 3[13] utilizando las bibliotecas Numeric Python (NumPy)[10] y Python Open Computing Language (PyOpenCL)[23]. Las consideraciones para esto fueron las siguientes:

Python 3 + Numpy[13, 10]: Esta elección es una de las más convenientes para computación científica debido a su alto nivel, facilidad de implementación, herramientas desarrolladas por sobre las mismas, disponibilidad de documentación. La eficiencia no se ve afectada por ser de alto nivel puesto que las operaciones matriciales están implementadas en C [22].

PyOpenCL[23, 17]: Esta biblioteca resultó muy útil puesto que permite paralelizar operaciones sobre arreglos bidimensionales, es decir, imágenes. El algoritmo de pareo de bloques está implementado como un kernel. Esto permite que los miles de bloques de cada imagen sean calculados en paralelo, opcionalmente en una GPU, que dispone de una gran cantidad de procesadores para este propósito.

4.2. Arquitectura

El diagrama de la arquitectura implementada es el de la figura 4.1. A continuación se entrará en detalle sobre las características y funciones de cada componente de la misma.

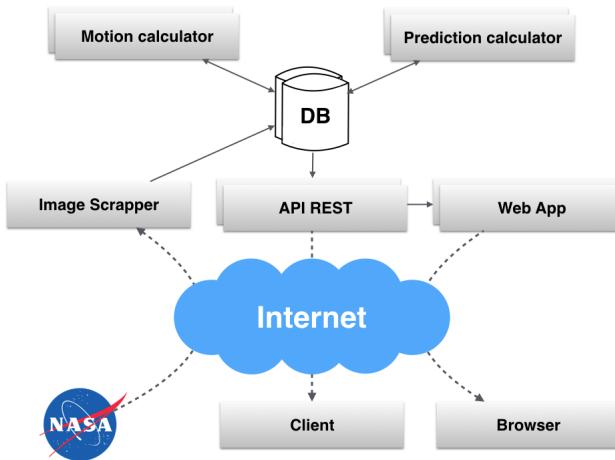


Figura 4.1: Arquitectura de la aplicación

4.2.1. Base de datos

Se utiliza una base de datos relacional para almacenar información tanto sobre los estados meteorológicos obtenidos como de algunos resultados parciales como el cálculo del movimiento para así poder obtener representaciones rápidamente.

El motor escogido es PostgreSQL [18] y es utilizado a través de SQLAlchemy [2] como ORM.

El esquema de la base de datos está representado en el diagrama UML de la figura 4.2. Se incluye adicionalmente un diagrama de entidad-relación (Figura 4.3) para una mejor comprensión del esquema.

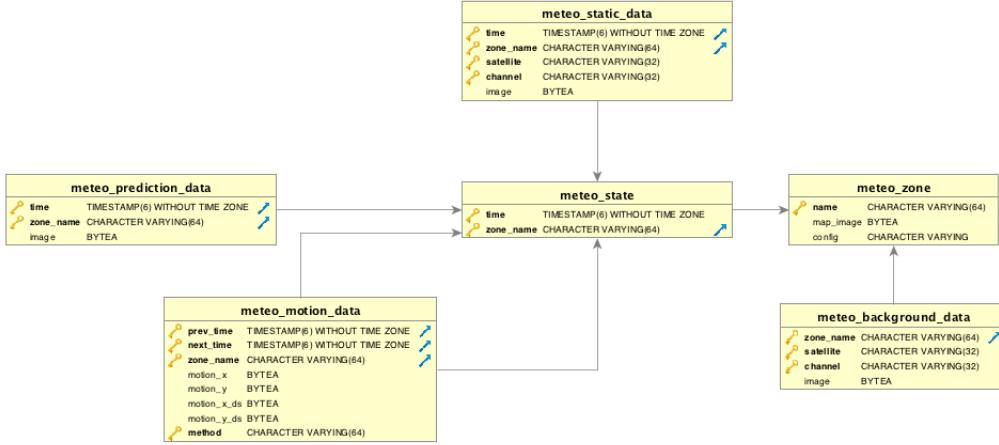


Figura 4.2: Diagrama UML del esquema de la base de datos.

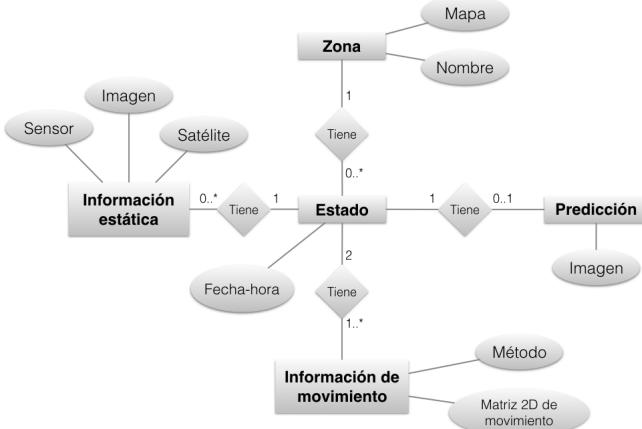


Figura 4.3: Diagrama de entidad-relación del esquema de la base de datos.

Este esquema se eligió con el objetivo de que permitiera analizar zonas por separado (funcionando independientemente y con su propia configuración), puesto que son entidades independientes. Por cada estado debe ser posible que haya más de una imagen satelital y una sola predicción. Finalmente se debe poder guardar información generada sobre el movimiento entre dos estados. Se decidió que las interpolaciones no estuvieran almacenadas en la base de datos porque al generarlas a demanda, la aplicación se vuelve más flexible (por ejemplo se puede elegir la cantidad de estados previos que se deseen, de lo contrario estaría delimitado).

4.2.2. Obtención de imágenes

La obtención de imágenes se realiza mediante un servicio simple hecho en Python que descarga periódicamente imágenes satelitales del servidor de GOES-NASA [25] por medio de FTP y/o HTTP. En particular, el satélite que se utiliza es el GOES13 por ser el que provee las imágenes de Argentina.

El mismo disponibiliza una nueva imagen cada aproximadamente 30 minutos. El servicio chequea en ese intervalo de tiempo si hay disponible alguna nueva imagen, en cuyo caso la descarga y la guarda en una base de datos para ser luego consumida por el resto de los servicios.

4.2.3. Servicios de cálculo: Movimiento y predicciones

Estos servicios realizan polling sobre la base de datos con un intervalo de aproximadamente un minuto, configurable, lo que no es demasiado dados los tiempos de aparición de nuevas imágenes, mencionados en 4.2.2.

Movimiento: Calcula los movimientos estimados a partir de un par de imágenes. Normalmente una previamente existente y una nueva. Consiste en la implementación del algoritmo especificada en la sección 4.1.

Predicciones: Calcula predicciones a partir de nuevos datos. Se detalla más al respecto en la sección 4.1.3.

4.2.4. API

Se desarrolló una interfaz REST/JSON [12, 21] para disponibilizar los distintos resultados a cualquier tipo de servicio que requiera consumirlos. En la misma se provee:

- Todas las imágenes recolectadas en distintos tipos de sensor, en formato PNG [4].
- Animaciones realizadas a partir de la concatenación de estas imágenes, en formato GIF [20].
- Mapas de las regiones analizadas.
- Capas de movimiento estimado a partir de un intervalo de tiempo deseado, en formato PNG [4] con transparencia.
- Animaciones del recorrido de puntos estimado a partir del cálculo del movimiento, en formato GIF [20], superpuestas al tipo de sensor que le fuere indicado.

4.2.5. Aplicación web

La forma provista para visualizar la información calculada por el backend es por medio de una aplicación web hecha en Java [7], en particular usando el framework Wicket [15] junto con Spring MVC [27]. La aplicación se comunica con el backend, consumiendo la API REST [12] que este expone, utilizando para ello otro framework: Java Jersey [8]. La misma es accesible públicamente en <http://clouds.it.itba.edu.ar/> y está desplegada en un servidor del ITBA utilizando Apache Tomcat [14].

Descripción

La aplicación consta de tres pantallas a las cuales se puede acceder utilizando el menú superior.

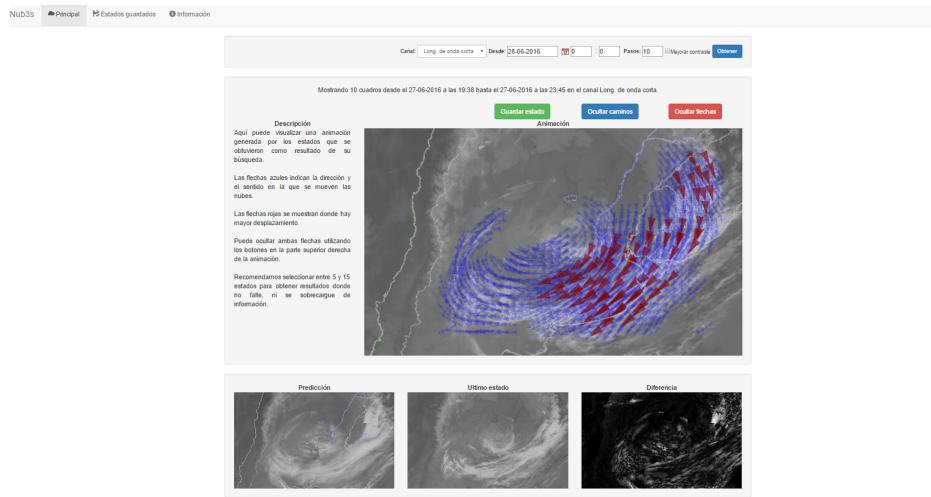


Figura 4.4: Aplicación web - Página principal.

En la pantalla ‘Principal’ (Figura 4.4) se pueden visualizar tres paneles: uno utilizado para filtrar el contenido de los paneles inferiores, otro donde se tiene una animación sobre la que se dibujan flechas azules y rojas -indicando la dirección y sentido de los vientos-, y por último se muestra el último estado de la selección, su predicción y la diferencia entre ambos, a fin de poder comparar qué tanto se ha acertado en la predicción.

A su vez el panel central muestra una descripción del significado de las distintas flechas junto con dos botones que permiten ocultarlas en caso de que se desee visualizar mejor la animación. Un tercer botón permite guardar la selección que se está observando en el momento actual, archivando el estado que está seleccionado en el momento para poder recuperarlo luego. Este estado se guarda en una cookie del explorador y puede ser recuperado accediendo a la pantalla de Estados guardados. En la cookie se guarda lo necesario para poder recuperar el estado guardado, es decir, la fecha y hora, el canal, la cantidad de estados y si se eligió mejorar el contraste o no.

Historial de estados guardados				
Fecha	Canal	Pasos	Mejora contraste	Opción
25-07-2016 23:45	Longitud de onda corta	10	No	Recuperar estado
03-07-2016 23:45	Longitud de onda corta	10	No	Recuperar estado
27-06-2016 23:45	Longitud de onda corta	10	No	Recuperar estado
14-06-2016 23:45	Longitud de onda corta	10	No	Recuperar estado

Figura 4.5: Aplicación web - Estados guardados.

En la pantalla de ‘Estados guardados’ (Figura 4.5) se lista la información de los estados que fueron guardados desde la pantalla Principal. Se puede observar la fecha y hora, el canal, la cantidad de pasos, si se eligió mejorar el contraste o no y por último un botón que permite recuperarlo. Al presionar este botón, volvemos a la pantalla principal, donde podremos ver el estado que ha sido guardado.

The screenshot shows a user interface for a project titled 'Nub3s'. At the top, there are navigation links: 'Principal', 'Estados guardados', and 'Información'. Below these, there are three main sections:

- Integrantes:** Lists team members: Diaz Uzbo, Pablo - pdiazubo@ib.edu.ar; Ell, Federico - felli@ib.edu.ar; Pomar, Federico - tpomar@ib.edu.ar.
- Tutor:** Lists the advisor: Ignacio Alvarez-Hamelin - ihameli@ib.edu.ar.
- Información adicional:** Contains two paragraphs of text describing the project's implementation and visual style, followed by a 'Descargar informe' button.

Figura 4.6: Aplicación web - Información sobre los integrantes.

La última pantalla de la aplicación (Figura 4.6) contiene información de contacto del equipo de desarrollo junto con un botón para descargar una copia de este informe en formato PDF.

Capítulo 5

Resultados

En este capítulo se mostrarán ejemplos de situaciones en las que la estimación del movimiento presenta problemas, así como otras en la que los resultados obtenidos son los esperados.

5.1. Problemas

La estimación del movimiento se ve afectada en las siguientes situaciones:

Eventos meteorológicos muy rápidos: Cuando aparecen sistemas de tormentas fuertes, con vientos rápidos donde ocurre fragmentación o agrupación repentina de nubes se pierde fidelidad en los resultados puesto que la frecuencia temporal es demasiado baja. Esto da lugar a que algunas de las correlaciones no reflejen el desplazamiento teórico esperado (Figura 5.1).

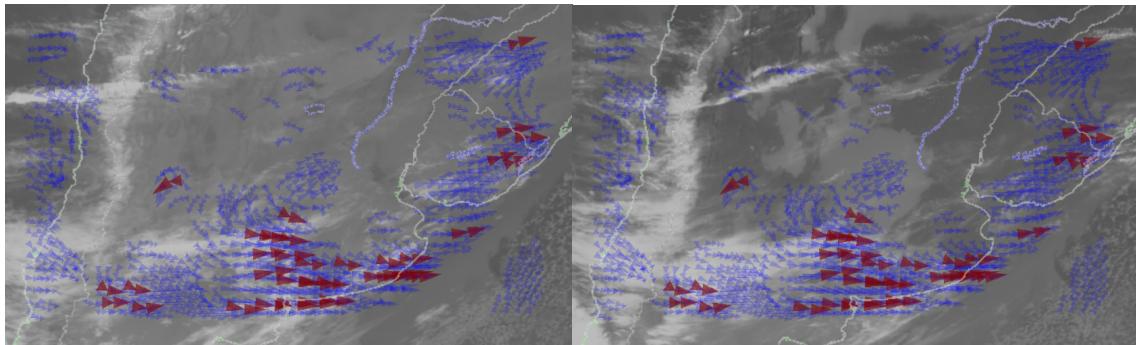


Figura 5.1: Por la rápida formación de la nube que se encuentra abajo a la izquierda, la dirección de algunas flechas no refleja la dirección real del movimiento.

Evaporación y precipitación: Tanto el método de Lucas y Kanade [24] como la correlación de características [1] suponen que se conserva la luminosidad. Esto puede no ocurrir en el caso de una evaporación o precipitación de agua en la atmósfera. En estas situaciones es común que se perciba movimiento donde no lo hay. Por ejemplo, es posible que al disminuir el tamaño de una nube, aparezca falso movimiento en los bordes, en el sentido de la disminución (Figura 5.2).

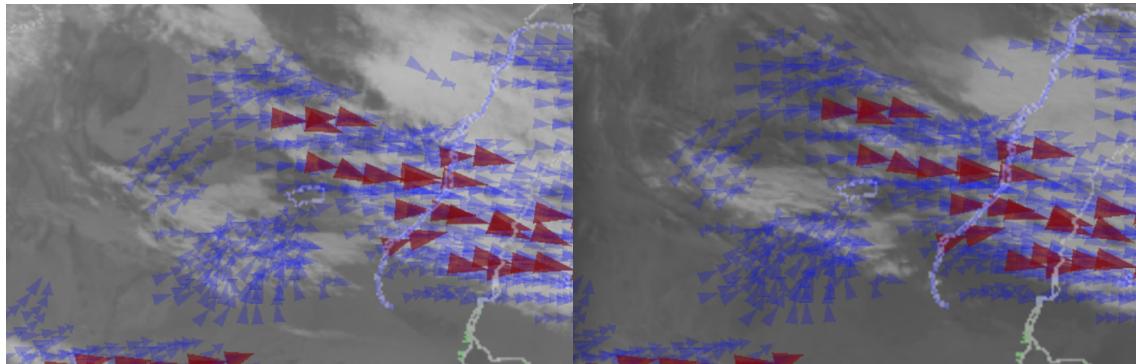


Figura 5.2: La disminución del tamaño de la nube de abajo a la izquierda es interpretada como movimiento.

Ciclo día-noche: Si bien se remueve el fondo de las imágenes, como se explicó en la subsección 4.1.1, en algunos canales, como IR2, sigue existiendo una variación de luminosidad homogénea debida a la reflexión de parte de la radiación solar durante el día. Ésta puede afectar la capacidad de detección puesto que el gradiente de la misma se desplaza a lo largo del tiempo y es tomado como un movimiento. (Figura 5.3).

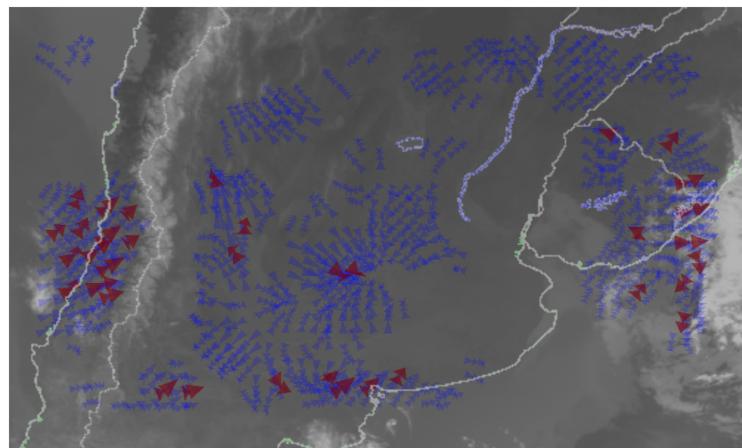


Figura 5.3: Por la rápida formación de la nube que se encuentra abajo a la izquierda, la dirección de algunas flechas no refleja la dirección real del movimiento.

Bordes de la imagen: Como es de esperar, por falta de información resulta imposible la detección del movimiento en los bordes de la imagen. Esto puede afectar a la hora de ajustar las trayectorias cerca de los extremos. (Figura 5.4)

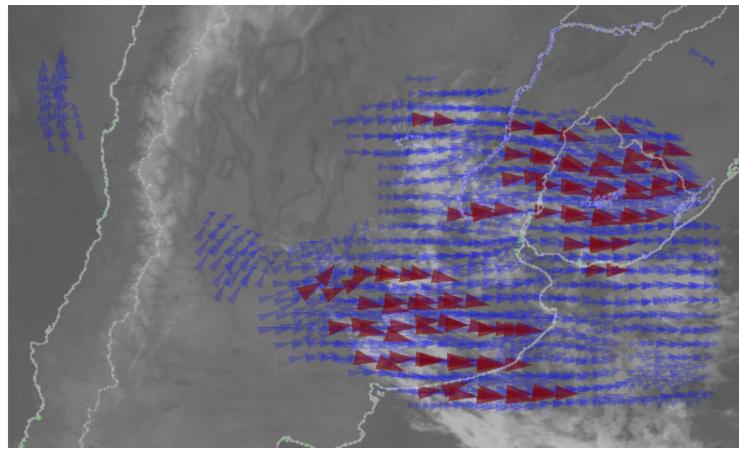


Figura 5.4: No existe información de movimiento en los bordes de la imagen.

5.2. Casos de éxito

El primer ejemplo de éxito donde la estimación del movimiento resultó ser la esperada se trata de un ciclón que fue captado por el satélite el día 28 de junio de 2016. En la Figura 5.5 puede observarse que la dirección de las flechas acompaña el movimiento circular descripto por las nubes entre la primera imagen y la segunda.

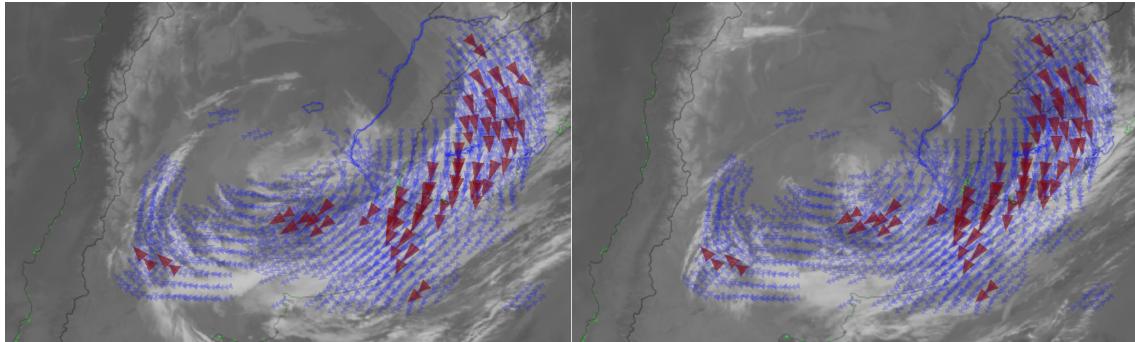


Figura 5.5: Ciclón detectado correctamente el día 28 de junio de 2016.

Otro ejemplo se encuentra el día 7 de agosto del 2016. En este caso, se pueden observar en la Figura 5.6 dos frentes de viento con distintas direcciones. Por un lado, un frente frío proveniente de la Cordillera de los Andes con dirección hacia el Este y, por el otro, uno cálido proveniente de Brasil con dirección hacia el Sur. Dadas estas condiciones, cuando ambos se encuentran la probabilidad de precipitación es alta y esto se pudo comprobar ya que el día 8 de agosto del 2016 llovió en la Ciudad de Buenos Aires.

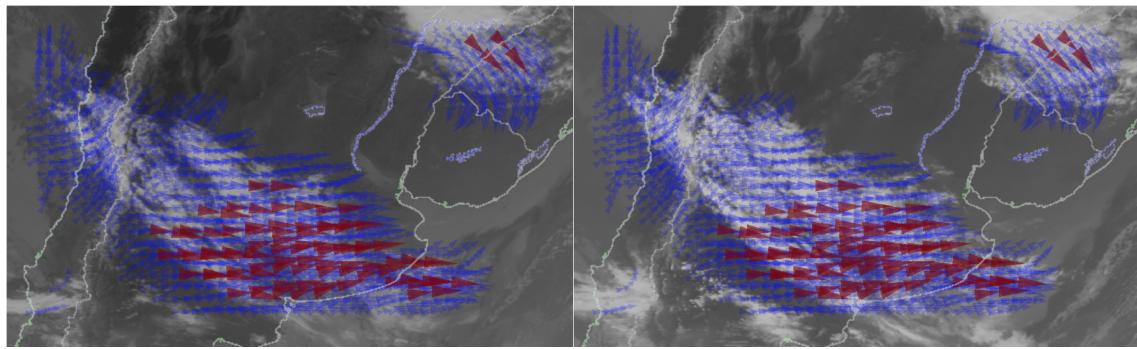


Figura 5.6: Secuencia en la que se observan dos frentes avanzar en simultáneo con distintas direcciones. Imágenes tomadas el 7 de agosto del 2016.

Capítulo 6

Conclusiones

En este capítulo se presentarán las conclusiones arribadas durante la realización de este trabajo, así como posibles futuras ampliaciones que resultan de interés.

6.1. Conclusiones generales

La meteorología es una ciencia que, debido a su carácter interdisciplinario y global y a la impredecibilidad de sus fenómenos, su estudio resulta de una complejidad elevada. Así mismo, la detección automática de movimiento en imágenes presenta también sus propias dificultades. Combinar estos dos aspectos, por lo tanto, no es una tarea sencilla.

En este trabajo se propuso utilizar una variante del algoritmo de pareo de bloques, presentado en la sección 3.2.4, para el seguimiento automático de nubes en una secuencia de imágenes satelitales con el fin de estimar la intensidad y dirección del viento. Los resultados obtenidos son alentadores e indican que, si bien esta implementación posee algunas limitaciones, como las mostradas en el capítulo 5, es una alternativa viable para este propósito. Sin embargo, si dicha secuencia no cuenta con un intervalo temporal entre imágenes relativamente bajo como en este caso (máximo una hora), los mismos podrían no ser muy buenos.

Otro de los objetivos propuestos por este proyecto consistió en proveer una predicción simple de la posición y tamaño de las nubes para las próximas horas. En la práctica, resultó no ser tan precisa como se esperaba en la mayoría de los casos. En la sección 6.2 se propone una forma de mejorarlala mediante la clasificación de los distintos tipos de nubes.

Por otro lado, la flexibilidad de la plataforma provista permite que futuras ampliaciones aprovechen los datos calculados por la misma con relativa facilidad. Para esto, el desarrollo de una API como método de interacción con el back-end resultó fundamental, ya que proporciona una forma de acceso simple e intuitiva a la información del mismo y permite, entre otras cosas, la utilización de front-ends alternativos. Este fue uno de los principales objetivos planteados al comienzo del proyecto.

6.2. Posibles ampliaciones

Entre las posibles ampliaciones de este trabajo podemos mencionar algunas que resultan de interés:

- Expandir la región geográfica abarcada por este proyecto a otras regiones del planeta e incluso a todo el mundo, para tener una visión más completa de cómo interactúan los sistemas de distintas regiones entre sí.
- Identificar y clasificar los diferentes tipos de nubes con el fin de brindar un pronóstico más preciso. Esto involucraría separar aquellas que tienen más probabilidades de precipitar y darles seguimiento para predecir donde y cuando lo harán.

- Detectar situaciones anómalas y ofrecer un sistema de alerta temprana frente a posibles catástrofes naturales o tormentas fuertes en las que podría precipitar, por ejemplo, granizo.
- Modelar el sistema de nubes como una red, para ver si su comportamiento es similar al de algún sistema conocido y tratarlo con técnicas utilizadas en este área.

Bibliografía

- [1] Aroh Barjatya. Block matching algorithms for motion estimation. *IEEE Transactions Evolution Computation*, 8(3):225–239, 2004.
- [2] Michael Bayer. Sqlalchemy. <http://www.sqlalchemy.org/>, jul 2016.
- [3] Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995.
- [4] Thomas Boutell. Png (portable network graphics) specification version 1.0. Technical report, 1997.
- [5] Remus Brad and Ioan Alfred Letia. Extracting cloud motion from satellite image sequences. In *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, volume 3, pages 1303–1307. IEEE, 2002.
- [6] Wayne Bresky and Jaime Daniels. The feasibility of an optical flow algorithm for estimating atmospheric motion. In *Proceedings of the Eighth Int. Winds Workshop, Beijing, China*, pages 24–28. Citeseer, 2006.
- [7] Oracle Corporation. Java. <http://www.java.com/>, jul 2016.
- [8] Oracle Corporation. Jersey. <https://jersey.java.net/>, jul 2016.
- [9] Ministerio de Defensa Argentino. Servicio meteorologico nacional. <http://www.smn.gov.ar/>, jul 2016.
- [10] Numpy Developers. Numpy. <http://www.numpy.org/>, jul 2016.
- [11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [12] Roy Fielding. Fielding dissertation: Chapter 5: Representational state transfer (rest). https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, jul 2016.
- [13] Python Software Foundation. Python. <https://www.python.org/>, jul 2016.
- [14] The Apache Software Foundation. Tomcat. <http://tomcat.apache.org/>, jul 2016.
- [15] The Apache Software Foundation. Wicket. <http://wicket.apache.org/>, jul 2016.
- [16] James J Gibson. The perception of the visual world. 1950.
- [17] Khronos Group. Opencl. <https://www.khronos.org/opencl/>, jul 2016.
- [18] The PostgreSQL Global Development Group. Postgresql. <https://www.postgresql.org/>, jul 2016.

- [19] Stephen J Huston and Holger G Krapp. Visuomotor transformation in the fly gaze stabilization system. *PLoS Biol*, 6(7):e173, 2008.
- [20] CompuServe Inc. Graphics interchange format. <https://www.w3.org/Graphics/GIF/spec-gif89a.txt>, jul 1990.
- [21] ECMA International. Json. <http://www.json.org/>, jul 2016.
- [22] Brian Kernighan and Dennis Ritchie. *The C Programming Language*. Prentice Hall, 1978.
- [23] Andreas Klockner. Pyopencl. <https://mathematician.de/software/pyopencl/>, jul 2016.
- [24] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- [25] NASA. Goes. <http://goes.gsfc.nasa.gov/>, jul 2016.
- [26] NullschoolEarth. Nullschoolearth. <https://earth.nullschool.net/>, jul 2016.
- [27] Pivotal Software. Spring. <https://spring.io/>, jul 2016.
- [28] Windyty. Windyty. <https://www.windyty.com/>, jul 2016.

Anexo