

# Métodos de análisis de imágenes aplicados al procesamiento de datos de una cámara SO<sub>2</sub>

---

Salvador Pedraza Espitia<sup>1</sup>

Tutor: Dr. Robin Campion

Marzo 2017

<sup>1</sup> Facultad de Ciencias  
Universidad Nacional Autónoma de México



# Contenido

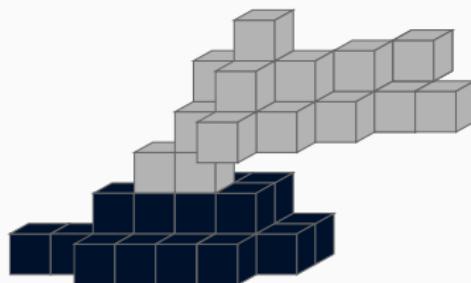
1. Introducción
2. Marco teórico
3. Metodología
4. Resultados
5. Conclusiones y recomendaciones

# Introducción

---

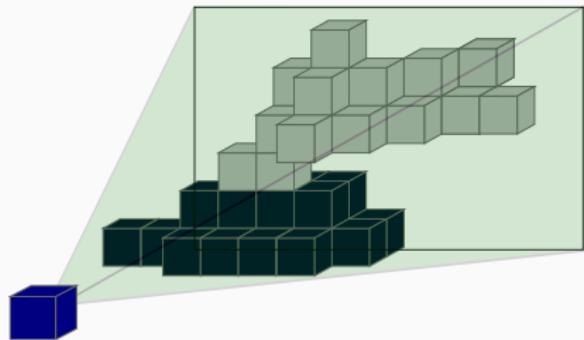
# Introducción

- Los procesos de desgasificación controlan la intensidad de las erupciones.
- Mediciones remotas permiten estudiar gases volcánicos.
- Una cámara SO<sub>2</sub> adquiere arreglos bidimensionales con frecuencia de hasta 1 Hz.



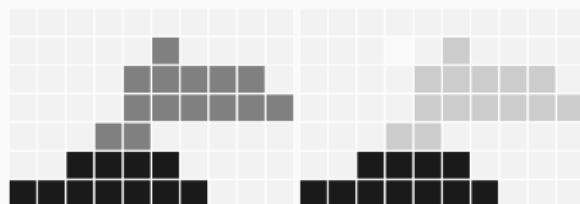
# Introducción

- Los procesos de desgasificación controlan la intensidad de las erupciones.
- Mediciones remotas permiten estudiar gases volcánicos.
- Una cámara SO<sub>2</sub> adquiere arreglos bidimensionales con frecuencia de hasta 1 Hz.



# Introducción

- Los procesos de desgasificación controlan la intensidad de las erupciones.
- Mediciones remotas permiten estudiar gases volcánicos.
- Una cámara SO<sub>2</sub> adquiere arreglos bidimensionales con frecuencia de hasta 1 Hz.



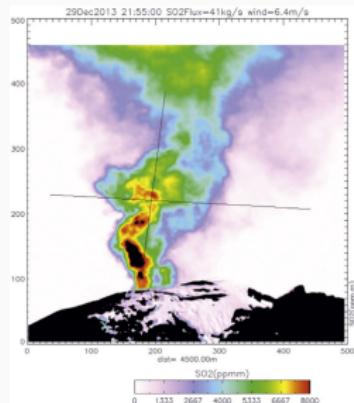
# Planteamiento del problema

- El procesamiento de imágenes para obtener concentraciones de SO<sub>2</sub>
  1. depende de la selección manual de regiones de pixeles.
- El cálculo del flujo de masa de SO<sub>2</sub>
  1. depende de las velocidades,
  2. las velocidades son heterogéneas.



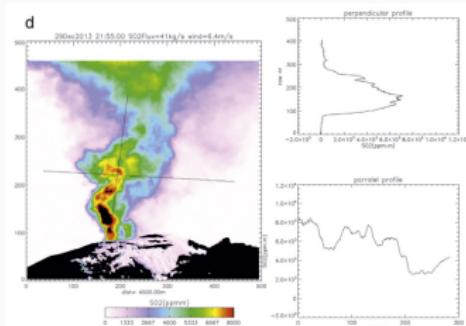
# Planteamiento del problema

- El procesamiento de imágenes para obtener concentraciones de SO<sub>2</sub>
  1. depende de la selección manual de regiones de pixeles.
- El cálculo del flujo de masa de SO<sub>2</sub>
  1. depende de las velocidades,
  2. las velocidades son heterogéneas.



## Planteamiento del problema

- El procesamiento de imágenes para obtener concentraciones de  $\text{SO}_2$ 
    1. depende de la selección manual de regiones de pixeles.
  - El cálculo del flujo de masa de  $\text{SO}_2$ 
    1. depende de las velocidades,
    2. las velocidades son heterogéneas.



# Hipótesis

- Las imágenes de la cámara SO<sub>2</sub> pueden usarse para extraer información que permite automatizar los algoritmos de procesamiento.

# Objetivos

- Programar en lenguaje PYTHON:
  1. correcciones de efectos causados por hardware
  2. detección automática de regiones del cielo y pluma
  3. correcciones de efectos de dilución de la radiación en la atmósfera.
  4. el cálculo de concentraciones de SO<sub>2</sub>
  5. el cálculo de campos de velocidades y flujo de masa de SO<sub>2</sub>

# Objetivos

- Programar en lenguaje PYTHON:
  1. correcciones de efectos causados por hardware
  2. detección automática de regiones del cielo y pluma
  3. correcciones de efectos de dilución de la radiación en la atmósfera.
  4. el cálculo de concentraciones de SO<sub>2</sub>
  5. el cálculo de campos de velocidades y flujo de masa de SO<sub>2</sub>

# Objetivos

- Programar en lenguaje PYTHON:
  1. correcciones de efectos causados por hardware
  2. **detección automática de regiones del cielo y pluma**
  3. correcciones de efectos de dilución de la radiación en la atmósfera.
  4. el cálculo de concentraciones de SO<sub>2</sub>
  5. el cálculo de campos de velocidades y flujo de masa de SO<sub>2</sub>

# Objetivos

- Programar en lenguaje PYTHON:
  1. correcciones de efectos causados por hardware
  2. detección automática de regiones del cielo y pluma
  3. correcciones de efectos de dilución de la radiación en la atmósfera.
  4. el cálculo de concentraciones de SO<sub>2</sub>
  5. el cálculo de campos de velocidades y flujo de masa de SO<sub>2</sub>

# Objetivos

- Programar en lenguaje PYTHON:
  1. correcciones de efectos causados por hardware
  2. detección automática de regiones del cielo y pluma
  3. correcciones de efectos de dilución de la radiación en la atmósfera.
  4. el cálculo de concentraciones de SO<sub>2</sub>
  5. el cálculo de campos de velocidades y flujo de masa de SO<sub>2</sub>

# Objetivos

- Programar en lenguaje PYTHON:
  1. correcciones de efectos causados por hardware
  2. detección automática de regiones del cielo y pluma
  3. correcciones de efectos de dilución de la radiación en la atmósfera.
  4. el cálculo de concentraciones de SO<sub>2</sub>
  5. el cálculo de campos de velocidades y flujo de masa de SO<sub>2</sub>

# Objetivos

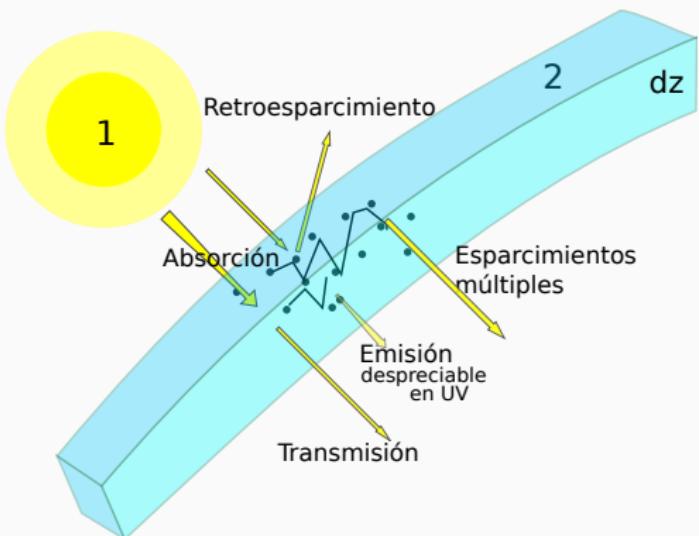
- Programar en lenguaje PYTHON:
  1. correcciones de efectos causados por hardware
  2. **detección automática de regiones del cielo y pluma**
  3. correcciones de efectos de dilución de la radiación en la atmósfera.
  4. el cálculo de concentraciones de SO<sub>2</sub>
  5. **el cálculo de campos de velocidades y flujo de masa de SO<sub>2</sub>**

## **Marco teórico**

---

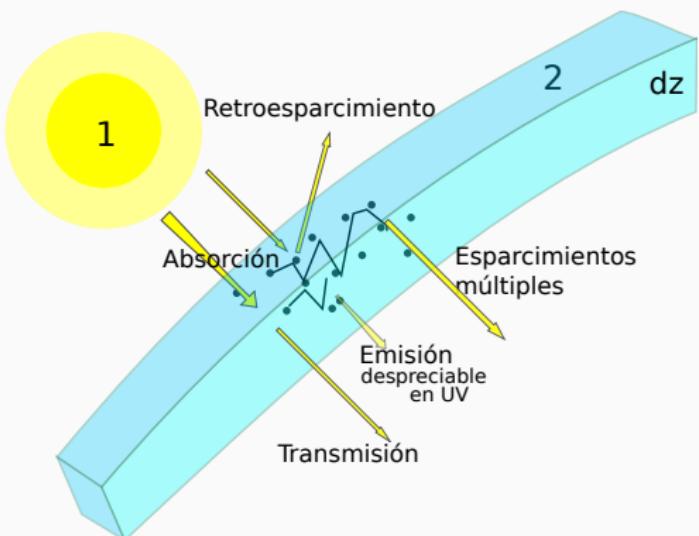
# Transferencia de radiación electromagnética en la atmósfera

- absorción  $\alpha$ ,
- esparcimiento  $\sigma$
- emisión



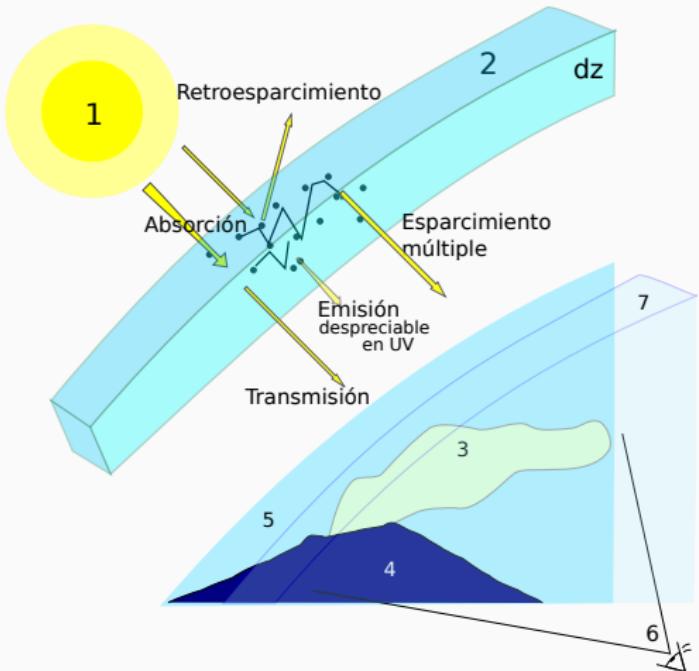
# Transferencia de radiación electromagnética en la atmósfera

- absorción  $\alpha$ ,
- **esparcimiento  $\sigma$**
- emisión



# Transferencia de radiación electromagnética en la atmósfera

- absorción  $\alpha$ ,
- esparcimiento  $\sigma$
- emisión



## Ecuación de la transferencia radiativa

$$\frac{dI_\lambda}{Nk_\lambda ds} = -I_\lambda + J_\lambda \quad (1)$$

$$\begin{aligned} \frac{dI_\lambda}{ds} &= -I_0 \left( \sum_i k_{\alpha,i} N_i + \beta_\alpha \right) \\ &\quad + (1 - \omega_\lambda) B_\lambda(T) \left( \sum_i k_{\alpha,i} N_i + \beta_\alpha \right) \\ &\quad + \left( \sum_i k_{\sigma,i} N_i + \beta_\sigma \right) \frac{\omega_\lambda}{4\pi} \int_{4\pi} I_\lambda(\hat{\Omega}') P_\lambda(\hat{\Omega}, \hat{\Omega}') d\Omega' \end{aligned} \quad (2)$$

## Ecuación de la transferencia radiativa

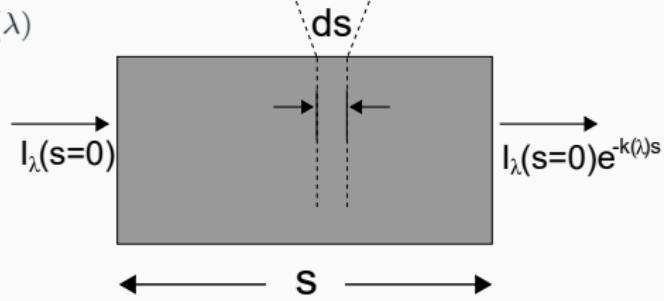
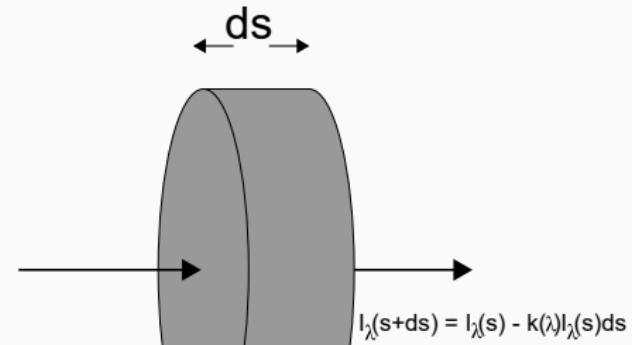
$$\frac{dI_\lambda}{Nk_\lambda ds} = -I_\lambda + J_\lambda \quad (3)$$

$$\begin{aligned} \frac{dI_\lambda}{ds} &= -I_0 \left( \sum_i k_{\alpha,i} N_i + \beta_\alpha \right) \\ &\quad + \left( \sum_i k_{\sigma,i} N_i + \beta_\sigma \right) \frac{\omega_\lambda}{4\pi} \int_{4\pi} I_\lambda(\hat{\Omega}') P_\lambda(\hat{\Omega}, \hat{\Omega}') d\Omega' \end{aligned} \quad (4)$$

# Extinción

- $dI_\lambda \propto -I_\lambda ds$
- Ley de Beer Lambert  
Bouguer

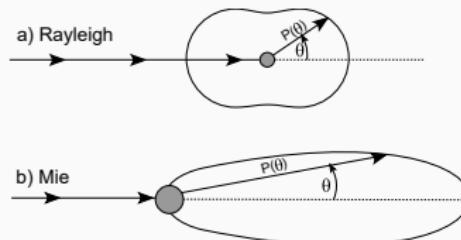
$$I_\lambda(s, \hat{\Omega}) = I_\lambda(0, \hat{\Omega}) e^{-\int_0^s ds' N k(\lambda)}$$



# Esparcimiento

- Rayleigh

$$I = I_0 \frac{9\pi^2}{2} \left( \frac{\epsilon - 1}{\epsilon + 2} \right)^2 (1 + \cos^2 \theta) \frac{nV^2}{\lambda^4 r^2} \quad (5)$$



- Mie

# Química atmosférica y gases volcánicos

Tabla: Principales gases que componen una atmósfera seca, libre de contaminantes. Platt y Stutz (2008)

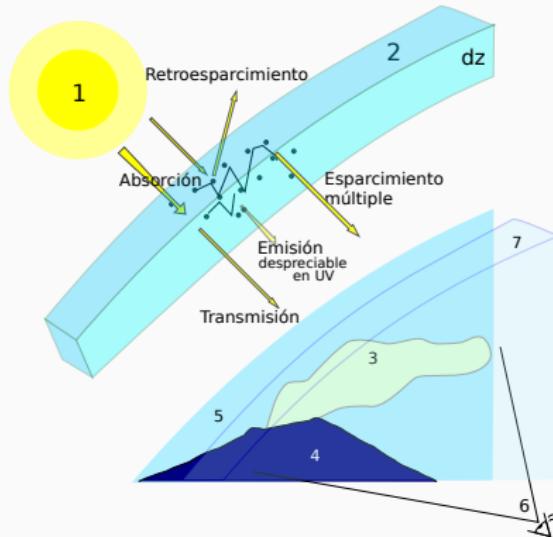
Gas	Formula química	Proporción por Vol [%]
Nitrógeno	N <sub>2</sub>	78.08
Oxígeno	O <sub>2</sub>	20.95
Argón	Ar	0.93
Dióxido de carbono	CO <sub>2</sub>	0.037
Neón	Ne	0.0018
Helio	He	0.00052
Metano	CH <sub>4</sub>	0.00017
Kriptón	Kr	0.00011
Xenón	Xe	0.00009
Hidrógeno	H <sub>2</sub>	0.00005
Dinitrógeno óxido	N <sub>2</sub> O	0.00003

Tabla: Principales gases característicos del gas magmático a su salida por el cráter. Textor et al. (2004)

Componente	Fórmula química	%/vol
Vapor de agua	H <sub>2</sub> O	50-90
Bióxido de carbono	CO <sub>2</sub>	1-40
Dióxido de azufre	SO <sub>2</sub>	1-25
Ácido sulfídrico	H <sub>2</sub> S	1-10

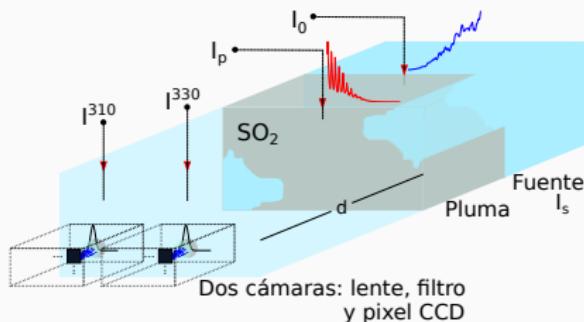
# Principio de medición de la cámara SO<sub>2</sub>

- La fuente de radiación es la radiación difusa en la atmósfera.
- Con frecuencias de hasta 1 Hz, se obtiene un par de imágenes simultáneas, una donde la absorción por moléculas de SO<sub>2</sub> es fuerte y la otra donde no.



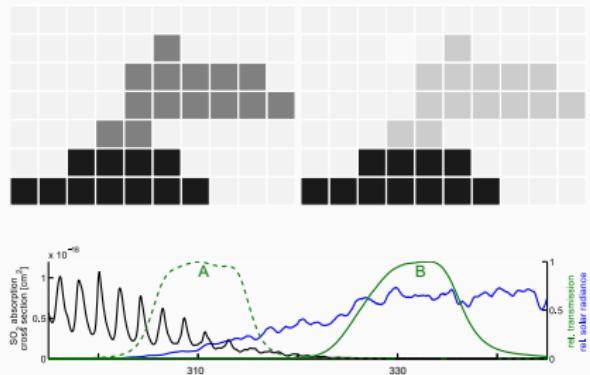
# Principio de medición de la cámara SO<sub>2</sub>

- La fuente de radiación es la radiación difusa en la atmósfera.
- Con frecuencias de hasta 1 Hz, se obtiene un par de imágenes simultáneas, una donde la absorción por moléculas de SO<sub>2</sub> es fuerte y la otra donde no.



# Principio de medición de la cámara SO<sub>2</sub>

- La fuente de radiación es la radiación difusa en la atmósfera.
- Con frecuencias de hasta 1 Hz, se obtiene un par de imágenes simultáneas, una donde la absorción por moléculas de SO<sub>2</sub> es fuerte y la otra donde no.



Adaptada de Kuhn *et al.* (2014)

# Principio de medición

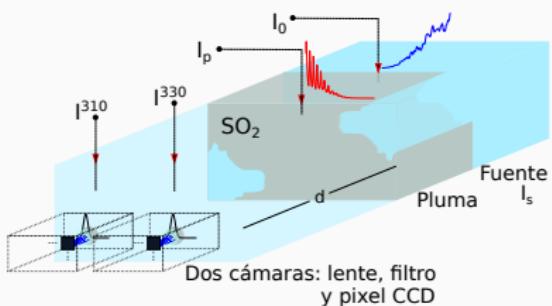
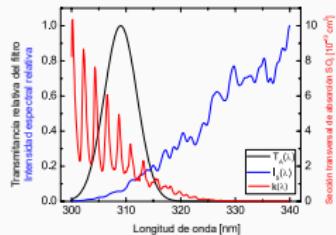
Radiación solar difusa libre de pluma  $I_0(\lambda) = I_S(\lambda)$

Beer Lambert Bouguer  $I_p(\lambda) = I_0(\lambda) \cdot e^{-k_\alpha(\lambda) \cdot S(\lambda)}$

Densidad de columna SO<sub>2</sub>  $S(\lambda) = \int_L N(x)dx$

Densidad óptica

$$\tau(\lambda) = -\ln \left( \frac{I_p(\lambda)}{I_0(\lambda)} \right) = k_\alpha(\lambda) \cdot S(\lambda) \quad (6)$$



Dos cámaras: lente, filtro  
y pixel CCD

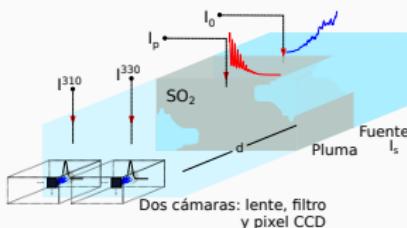
# Principio de medición de la cámara SO<sub>2</sub>

Radiación solar difusa, libre de pluma, medida por la cámara

$$I_0^{310} = \int_{\lambda} I_S(\lambda) \cdot T(\lambda) \cdot Q(\lambda) \, d\lambda \quad (7)$$

Radiación solar difusa que atraviesa la pluma, medida por la cámara

$$I_p^{310} = \int_{\lambda} I_S(\lambda) \cdot T(\lambda) \cdot Q(\lambda) \cdot e^{-k_{\alpha}(\lambda) \cdot S(\lambda)} \, d\lambda \quad (8)$$



## Principio de medición

No hay información espectral

$$\Rightarrow \hat{\tau} = -\ln \left( \frac{I_p(\lambda)}{I_0(\lambda)} \right) \quad (9)$$

$$\hat{\tau} = -\ln \left( \frac{I_0 \cdot \exp(-k_\alpha \cdot S)}{I_0} \right) = k_\alpha \cdot S \quad (10)$$

$$\Rightarrow \tau \approx k_\alpha \cdot S \quad (11)$$

## Principio de medición

No hay información espectral

$$\Rightarrow \hat{\tau} = -\ln \left( \frac{I_p(\lambda)}{I_0(\lambda)} \right) \quad (9)$$

$$\hat{\tau} = -\ln \left( \frac{I_0 \cdot \exp(-k_\alpha \cdot S)}{I_0} \right) = k_\alpha \cdot S \quad (10)$$

$$\Rightarrow \tau \approx k_\alpha \cdot S \quad (11)$$

## Principio de medición

$$\hat{\tau}^{310} = -\ln \left( \frac{I_p^{310}}{I_0^{310}} \right); \quad \hat{\tau}^{330} = -\ln \left( \frac{I_p^{330}}{I_0^{330}} \right)$$

Absorbancia Aparente

$$AA = \ln \left( \frac{I_0^{310}}{I_p^{310}} \middle/ \frac{I_0^{330}}{I_p^{330}} \right) = \hat{\tau}^{310} - \hat{\tau}^{330} \quad (12)$$

# Principio de medición

Absorbancia Aparente

$$AA = \ln \left( \frac{I_p^{330}}{I_p^{310}} \right) - \ln \left( \frac{I_0^{330}}{I_0^{310}} \right) \quad (13)$$

Para un arreglo 2D  $\Rightarrow \mathbf{I} = (I_{ij})$

$$\mathbf{AA} = \ln \left( \frac{\mathbf{I}_p^{330}}{\mathbf{I}_p^{310}} \right) - \ln \mathbf{B} \quad (14)$$

## Metodología

---

# Python



# Características de las imágenes de la cámara

- $512 \times 512$  pixeles con valores de 0 a  $2^{16} - 1$

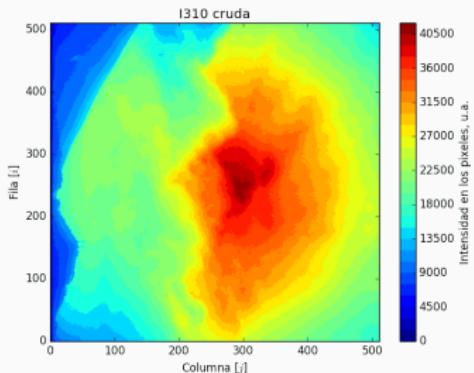


Imagen I310 cruda.

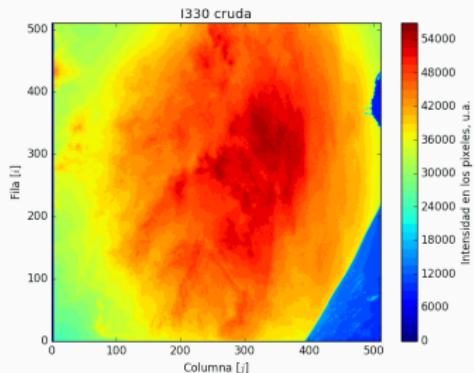
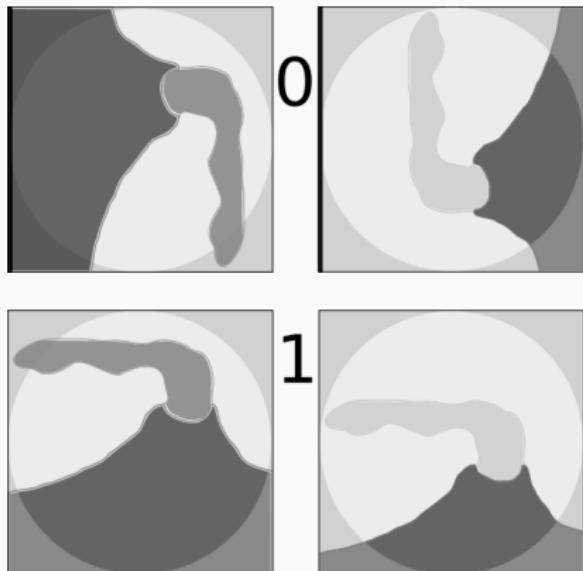


Imagen I330 cruda.

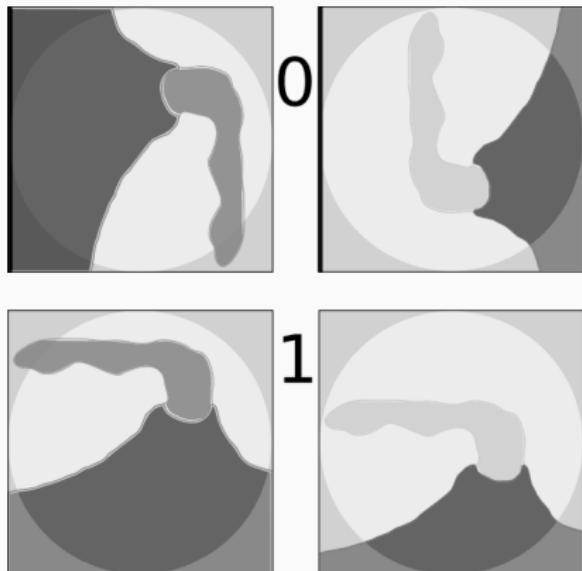
# Preparación de imágenes

- Eliminar columnas de pixeles defectuosos y bias.
- Orientar las imágenes en el mismo sentido.
- Vignetting.
- Desplazamiento por desalineación de cámaras.



# Preparación de imágenes

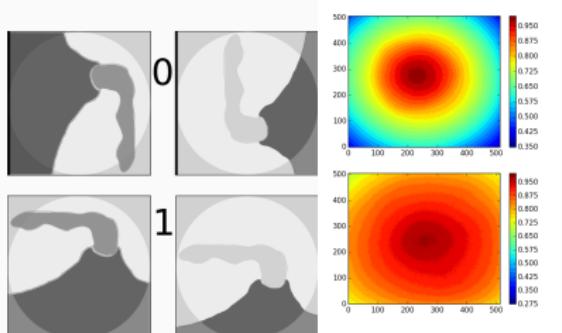
- Eliminar columnas de pixeles defectuosos y bias.
- Orientar las imágenes en el mismo sentido.
- Vignetting.
- Desplazamiento por desalineación de cámaras.



# Preparación de imágenes

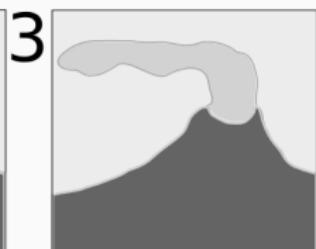
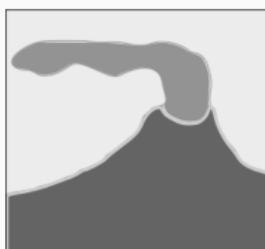
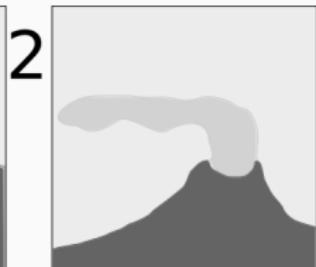
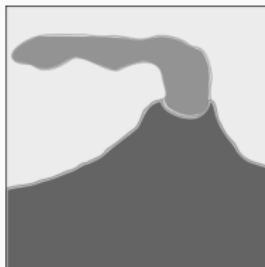
- Eliminar columnas de píxeles defectuosos y bias.
- Orientar las imágenes en el mismo sentido.
- Vignetting.
- Desplazamiento por desalineación de cámaras.

$$M = \frac{M_{\text{vignett}} \cdot \max(M_{\text{avg}})}{M_{\text{avg}}} = \frac{M_{\text{vignett}}}{\hat{M}_{\text{avg}}}$$

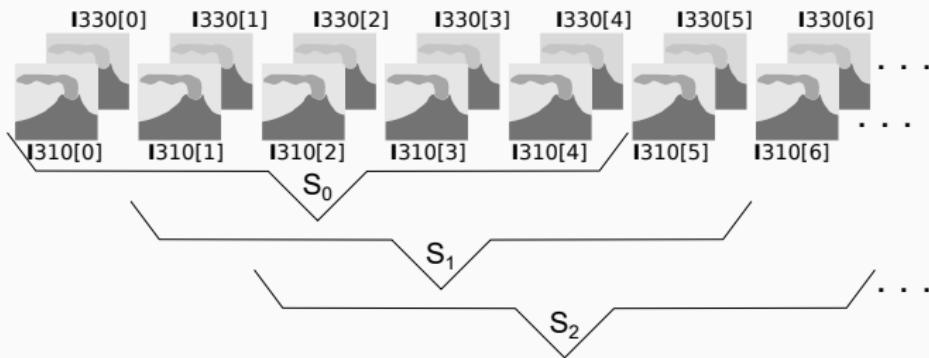


# Preparación de imágenes

- Eliminar columnas de pixeles defectuosos y bias.
- Orientar las imágenes en el mismo sentido.
- Vignetting.
- Desplazamiento por desalineación de cámaras.



# Secuencias



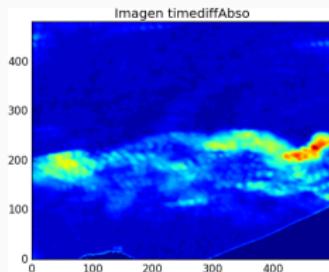
- $I^{310} = \{I^{310}[k] | k = 0, 1, 2, 3, 4\}$
- $I^{330} = \{I^{330}[k] | k = 0, 1, 2, 3, 4\}$

$$\mathbf{A} = \left\{ \mathbf{A}[k] = \log_{10} \frac{\mathbf{I}^{330}[k]}{\mathbf{I}^{310}[k]} \mid k=0,1,2,3,4 \right\} \quad (15)$$

# Imágenes artificiales

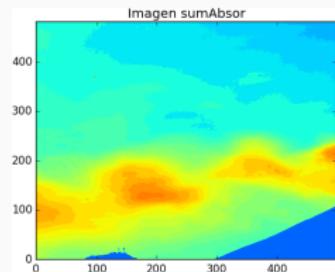
Detección de movimiento

$$\sum_{i=0}^3 |\mathbf{A}[i] - \mathbf{A}[i+1]|$$



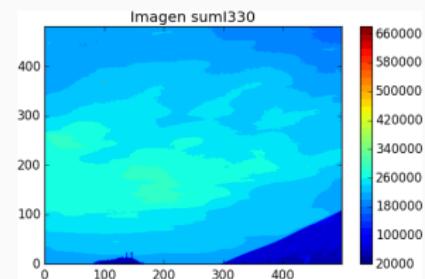
Resalta absorbancia

$$\sum_{i=0}^4 \mathbf{A}[i]$$



Cielo y volcán diferenciables

$$\sum_{i=0}^4 \mathbf{I}^{330}[i]$$



# Región de cielo recuperado

Histograma  $\left( \text{ } \right) =$

Histograma  $\left( \text{sumaAbso} * \text{ } \right) =$

Histograma  $\left( \text{timedifAbso} * \text{ } \right) =$

1 1 1	0 0 0	0 0 0	1*0*0 1*0*0 1*0*0	0 0 0
1 0 0	1 1 1	1 1 1	1*1*1 0*1*1 0*1*1	1 0 0
0 0 0	1 1 1	1 1 1	0*1*1 0*1*1 0*1*1	0 0 0

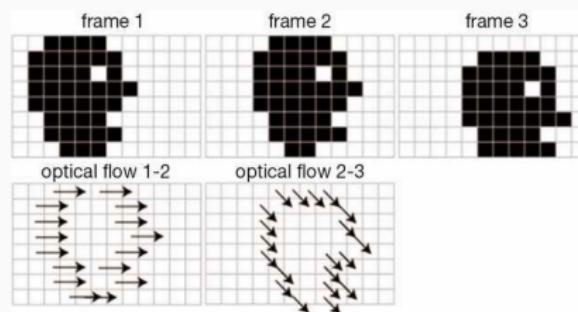


sumaimgs330 \* sumaAbso \* timedifAbso  $=$  regionCielo

# Flujo óptico

- Dado un conjunto de puntos en una imagen, encuentra esos mismos puntos en otra imagen.

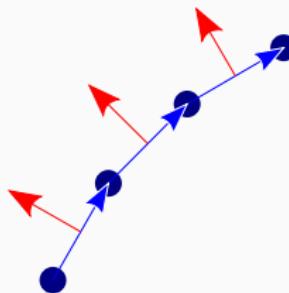
$$I_1(x, y) - I_2(x + \delta_x, y + \delta_y) \quad (16)$$



# Flujo de Masa SO<sub>2</sub>

$$\mathbf{AA} = \ln \left( \frac{\mathbf{I}_p^{330}}{\mathbf{I}_p^{310}} \right) - \ln \mathbf{B}$$

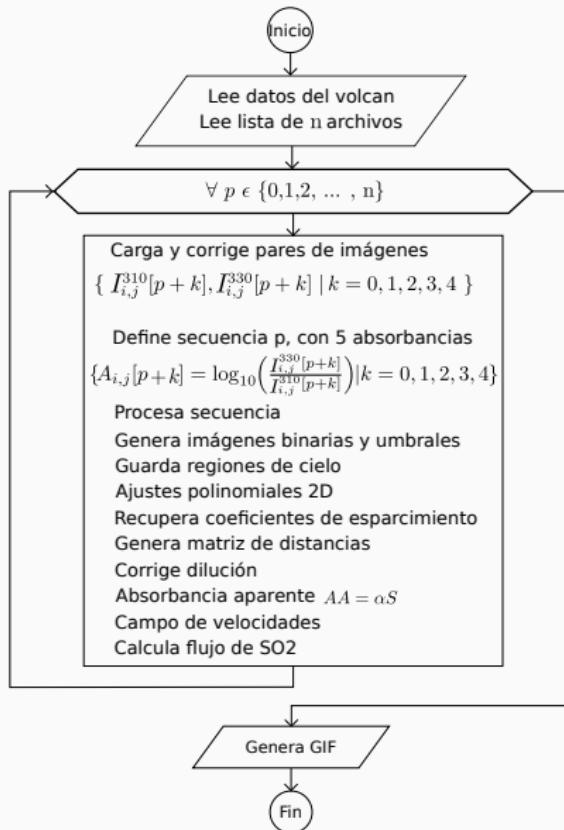
$$\int_{\Gamma} \vec{\mu} \cdot \vec{n} d\Gamma = \int \int_W \nabla \vec{\mu} dV \quad (17)$$



```
perp_vel = numpy.dot(velocities, int_norms.T)
```

```
fluxes = pix_vals * perp_vel * length_of_segments
```

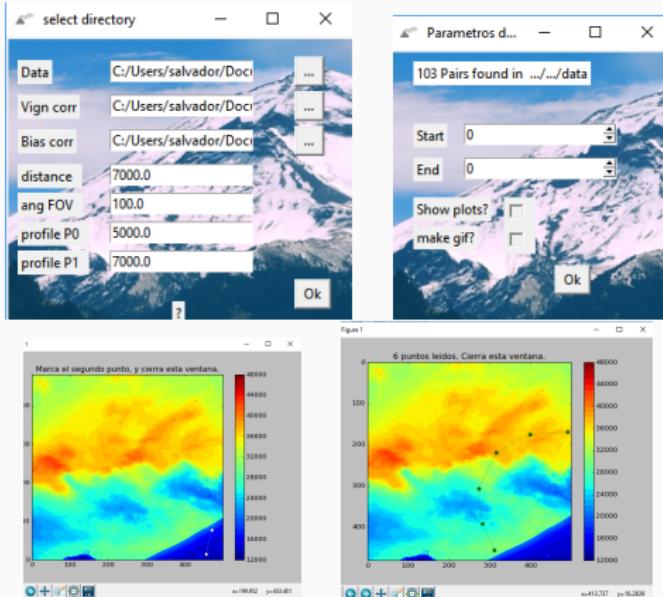
# Diagrama de flujo general



## **Resultados**

---

# Resultados (interfaz del programa)



# Imágenes preparadas

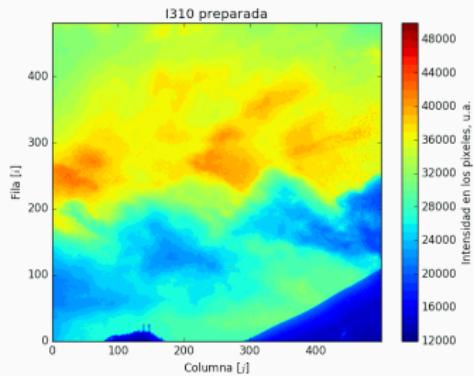


Imagen I310 Preparada.

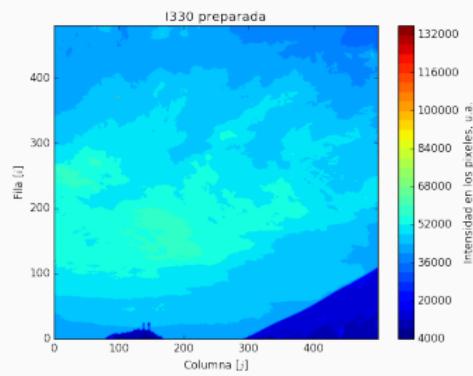
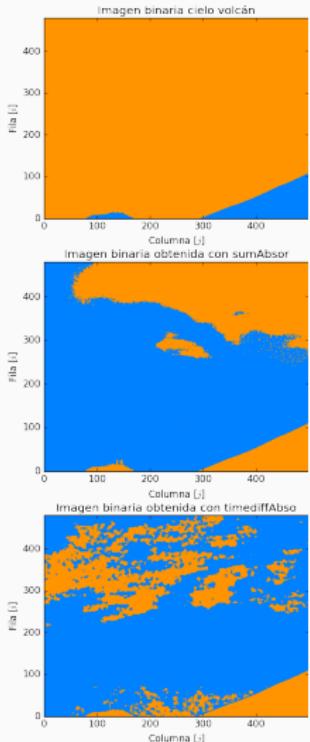
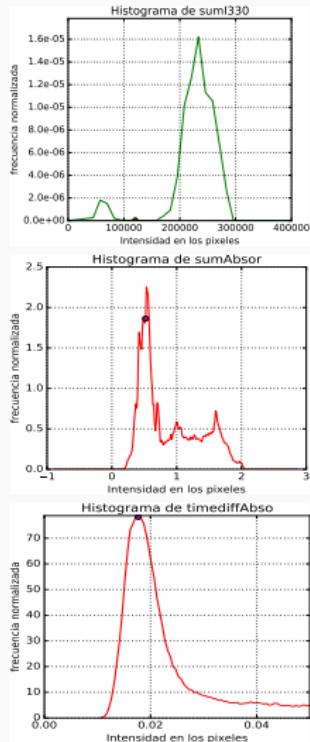
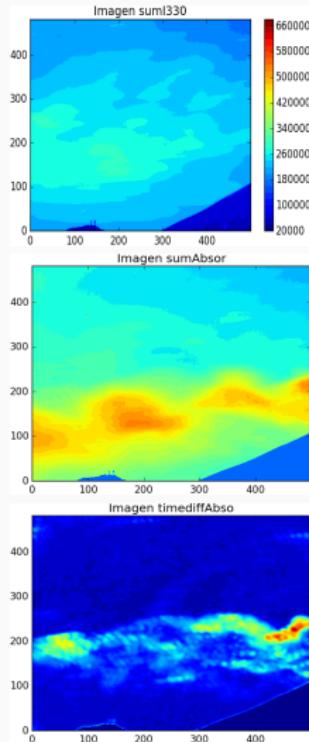


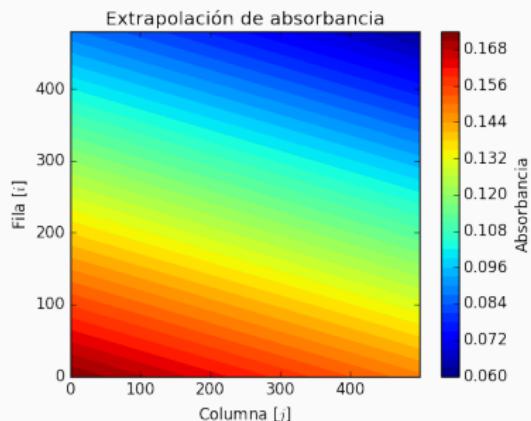
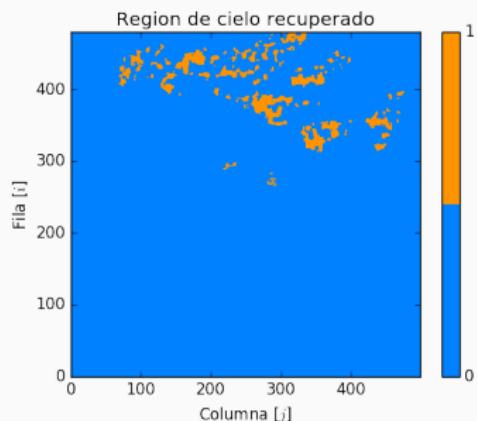
Imagen I330 Preparada.

# Popocatépetl

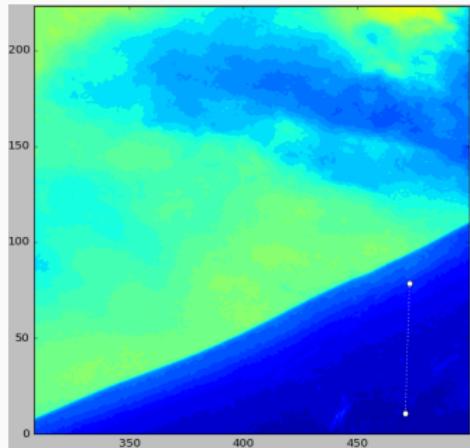
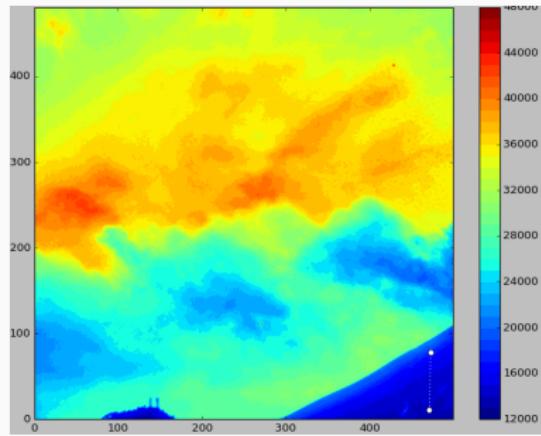


# Región de cielo recuperado

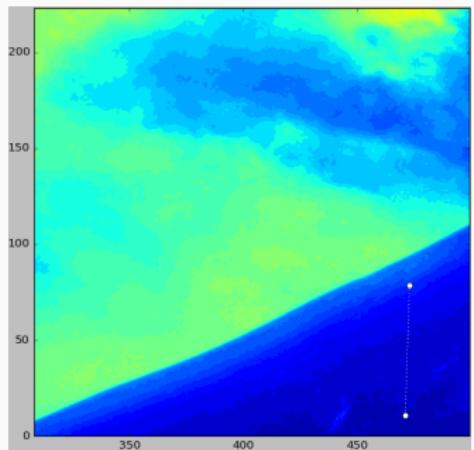
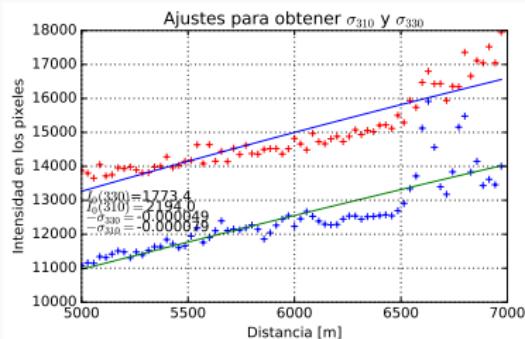
$$\mathbf{AA} = \ln \left( \frac{\mathbf{I}_p^{330}}{\mathbf{I}_p^{310}} \right) - \ln \mathbf{B}$$



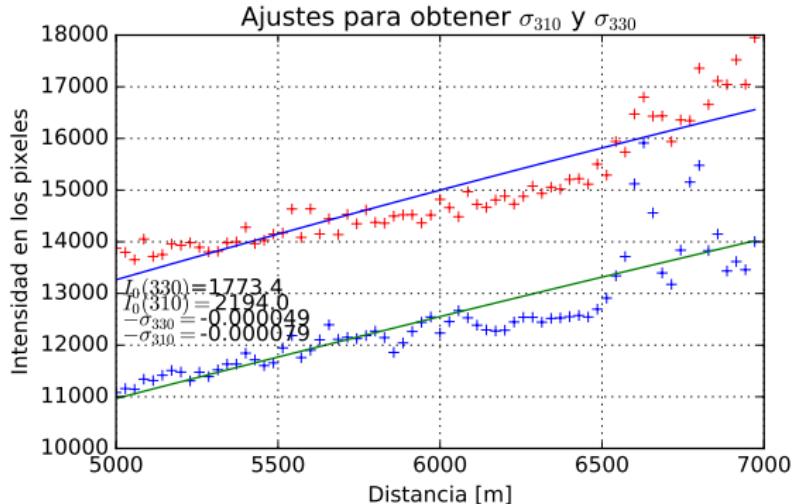
# Perfil para calcular dilución



# Perfil para calcular dilución



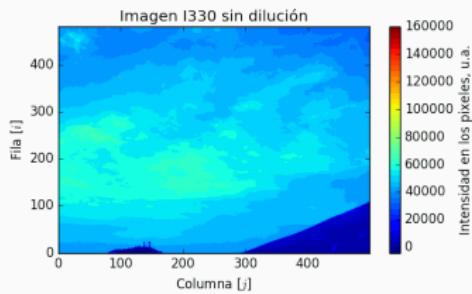
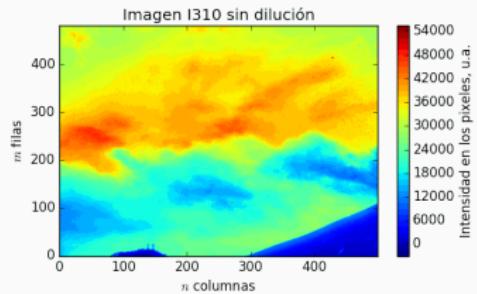
# Ajustes dilución



$$I(\lambda) = I_0(\lambda) \left( e^{-\sigma(\lambda)d} \right) + I_A(\lambda) \left( 1 - e^{-\sigma(\lambda)d} \right) \quad (18)$$

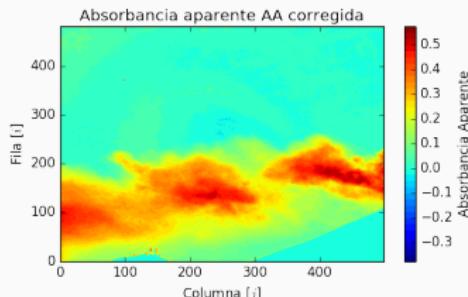
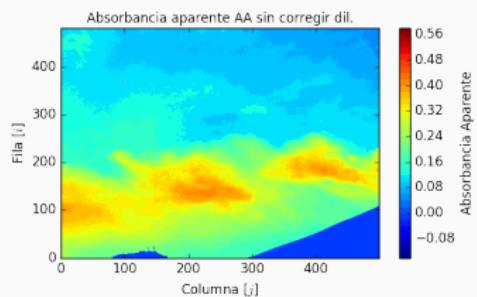
$$I_{\text{corr}_{i,j}} = \frac{I_{\text{uncorr}_{i,j}} - I_{A_{i,j}}(1 - e^{-\sigma d_{i,j}})}{e^{-\sigma d_{i,j}}} \quad (19)$$

# Imagenes sin dilución

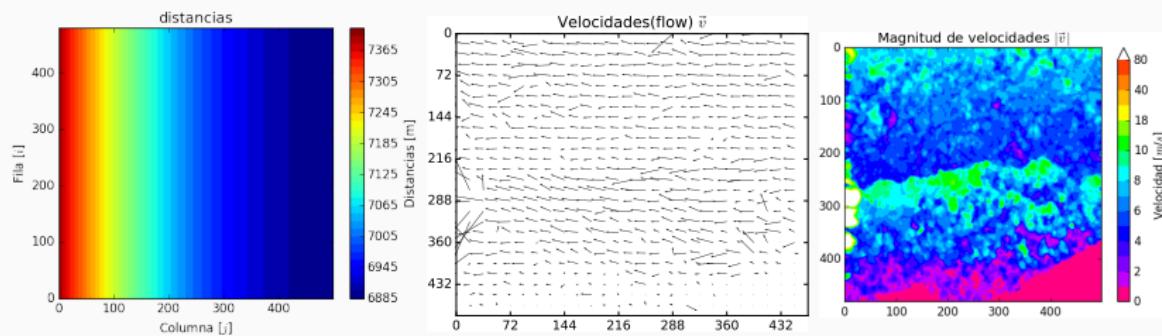


# Absorbancia aparente

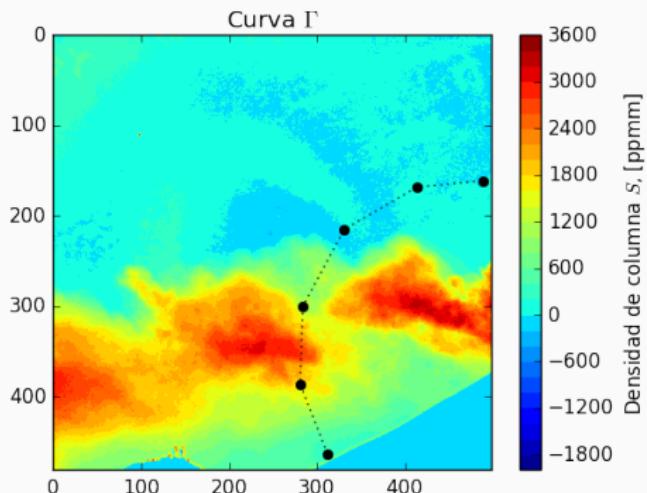
$$\mathbf{AA} = \ln \left( \frac{\mathbf{I}_p^{330}}{\mathbf{I}_p^{310}} \right) - \ln \mathbf{B}$$



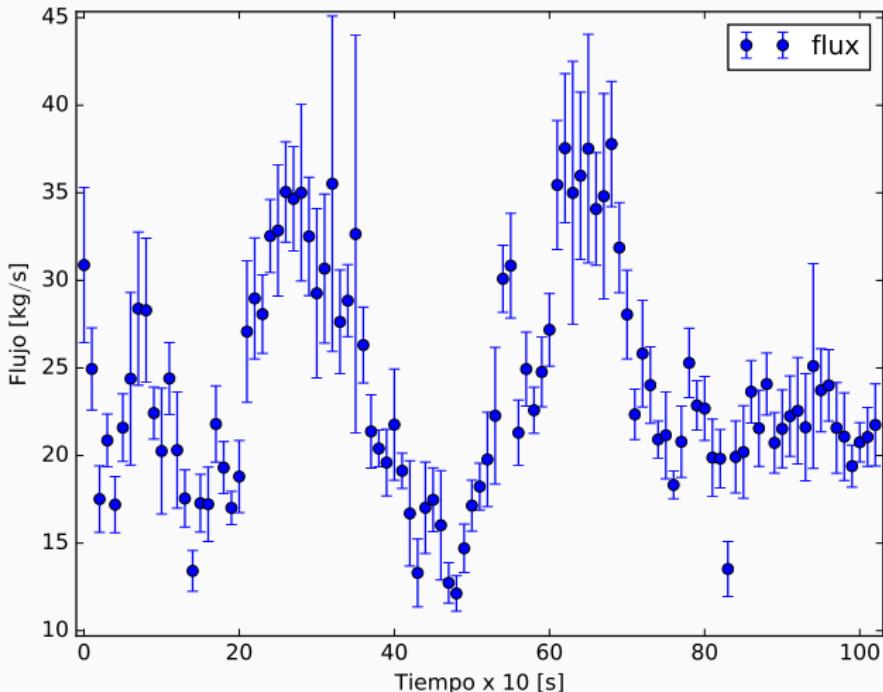
# Campo de velocidades



# Flujo de SO<sub>2</sub> en Popocatépetl

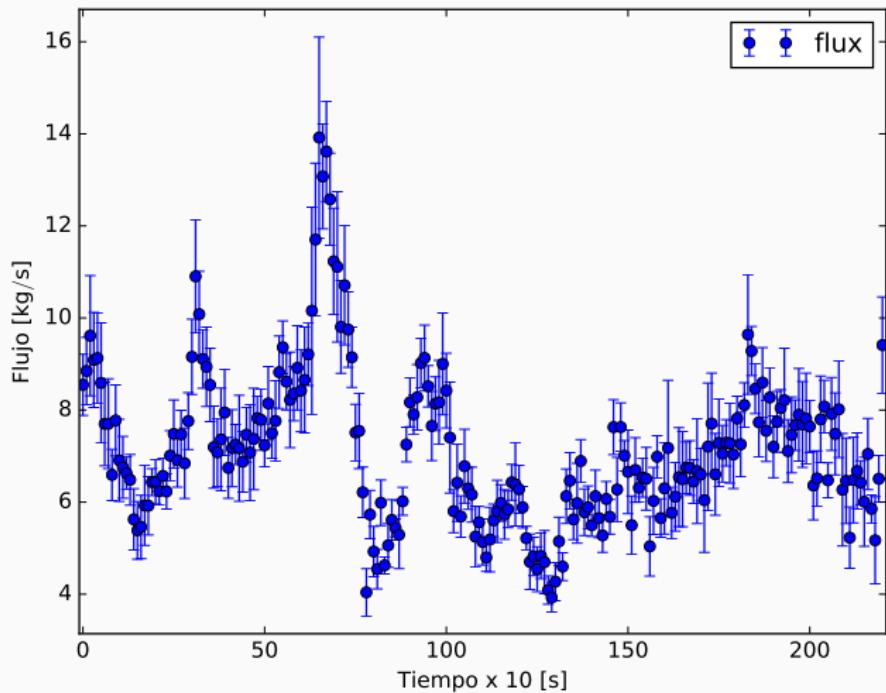


# Flujo de SO<sub>2</sub> en Popocatépetl



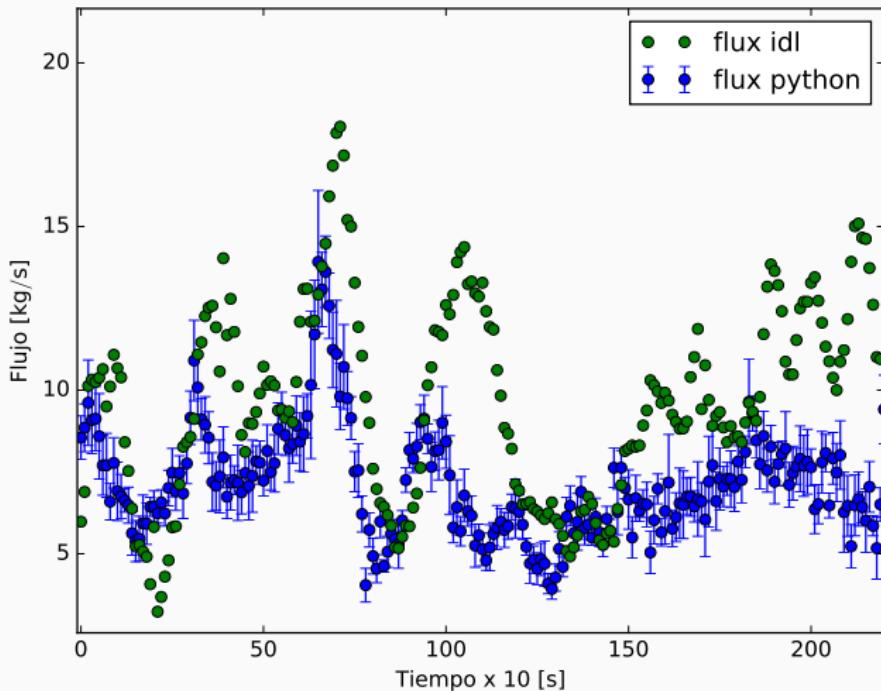
$t = 0 \Rightarrow 31/01/2013$  10 A.M. Hora local.

# Flujo de SO<sub>2</sub> en Ubinas



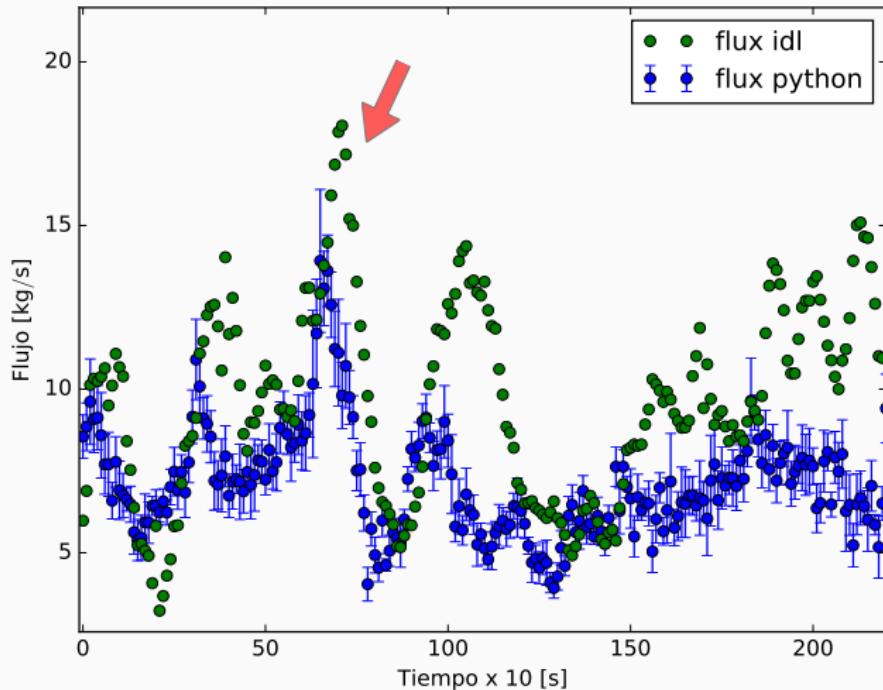
$t = 0 \Rightarrow 08/07/2015 19 : 25 : 10$  GMT.

# Flujo Ubinas



$t = 0 \Rightarrow 08/07/2015\ 19:25:10\ \text{GMT}.$

# Flujo Ubinas



$t = 0 \Rightarrow 08/07/2015 19 : 25 : 10$  GMT.

## **Conclusiones y recomendaciones**

---

## Conclusión i

- Importante monitorear gases volcánicos y entender los procesos de desgasificación.
- Conocer el principio físico de medición, de la cámara, permite diseñar los algoritmos de procesamiento.
- Detección automática de regiones de cielo para modelar radiación difusa detrás de la pluma (análisis de histogramas, morfología de imágenes binarias, extrapolación 2D, ...).
- Se hicieron correcciones a efectos de aerosoles, se recuperaron coeficientes de esparcimiento. Se logró obtener la distribución de columnas de  $\text{SO}_2$  usando la ley de Beer–Lambert–Bouguer.

## Conclusión ii

- Campo de velocidades usando un algoritmo de computación visual para obtener flujo óptico (operaciones matriciales – vectoriales).
- Flujo de masa de SO<sub>2</sub> utilizando el campo de velocidades y distribución 2D de columnas SO<sub>2</sub> y teorema de Gauss.
- Programa en Python versátil y con licencia de código abierto. Probado en mediciones de distintos volcán.
- Se procesaron imágenes del volcán Popocatépetl (hasta  $\approx 40$  kg/s), Ubinas (en promedio 8 kg/s), Pacaya ( $\approx 4$  kg/s).

## Recomendaciones

- Ajustar parámetros del cálculo del flujo óptico, cuidando que las deformaciones en la pluma volcánica sean aceptables (desplazamientos menores  $\sim 10\text{px}$ ).
- Integrar el programa de procesamiento con el programa de adquisición de imágenes y planear una estación fija de monitoreo.
- Agregar funcionalidades al programa para agregar mas de una línea de integración y más perfiles que permitan recuperar coeficientes de esparcimiento confiables.
- Probar otros algoritmos de detección automática, Sci-Kit de Python (Learning Machine).
- Comparación con datos sísmicos. Análisis de series de tiempo y detección de patrones de comportamiento.

## Apéndice A

---



# Contenido

## 1. Introducción

Planteamiento del problema  
Objetivos

## 2. Marco teórico

Ec transferencia radiativa  
Ley Beer Lambert

Principio de medición cámara  
 $\text{SO}_2$

## 3. Metodología

Diagrama de flujo

## 4. Resultados

Campo de velocidades  
Flujo de  $\text{SO}_2$

## 5. Conclusiones y recomendaciones