

Detección casera de radiación ambiental mediante cámaras digitales de consumo. Aplicaciones colaborativas

Francisco Albiol – Alberto Corbi

Resumen

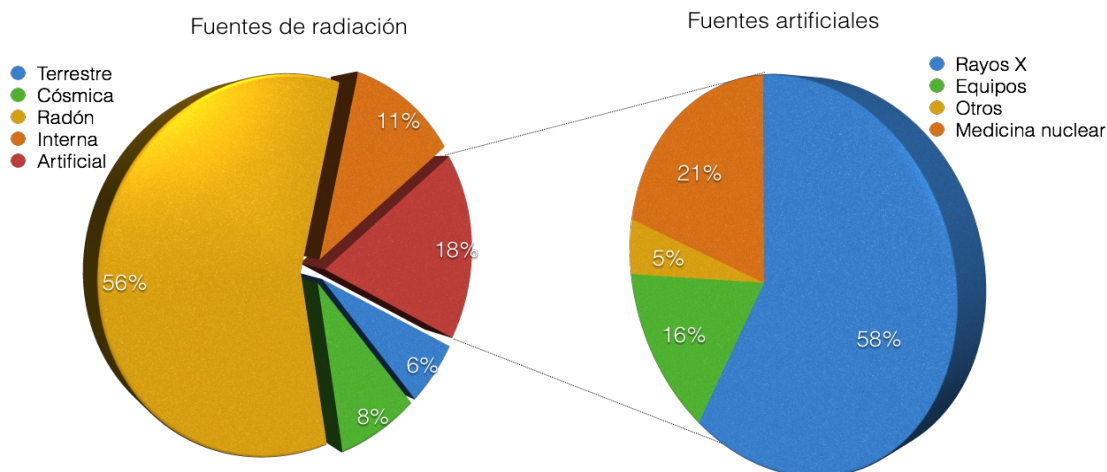
En este trabajo de divulgación presentamos un **entorno experimental** para el uso de dispositivos de **captura de imagen digital** como medio de detección casera de **radiación ambiental y cósmica**. Se trata de un conjunto cohesionado de herramientas de software completamente funcional sobre la gran mayoría de plataformas informáticas habituales (móvil, escritorio, Windows, Mac, Linux, Android e iOS) y de las correspondientes instrucciones básicas para su despliegue en todas ellas. A continuación detallamos estas herramientas, bibliotecas y funcionamiento básico del cliente de ejemplo que adjuntamos con la presente memoria.

Introducción

A continuación vamos a recordar las principales fuentes de radiación que nuestro dispositivo experimental puede detectar y los fundamentos físicos y tecnológicos sobre los que se basa nuestro proyecto.

La radiación ambiental

La radiación ambiental se encuentra de forma cotidiana en la naturaleza. Puede tener **distintos orígenes**, tanto terrestres como externos y tanto naturales como artificiales. Además puede ser de distintos tipos (según el tipo de partícula incidente y la energía depositada). Los seres humanos somos también, de manera casi inevitable, *dianas* de estas fuentes de radiación y llegamos a recibir, de media, 2.5 mSv por persona y por año, aunque en algunos países esta cantidad de radiación acumulada puede llegar hasta los 7 mSv. La exposición aumenta considerablemente en los países industrializados y tiene especial incidencia en entornos relacionados con el uso de la imagen médica.

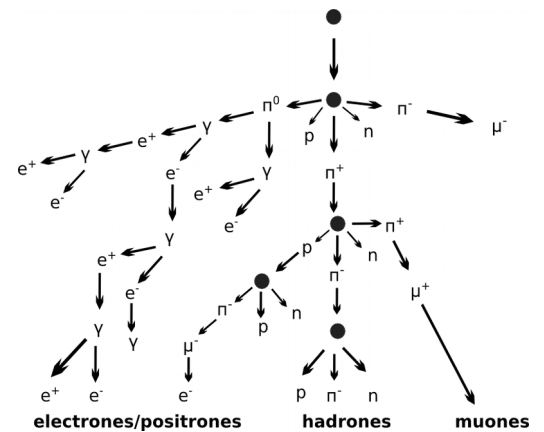


Ciertas instituciones y grupos profesionales cuentan con **detectores de radiación** tanto a nivel grupal como individual. Los sistemas de seguridad individual suelen ser sobretodo de carácter acumulativo y pueden poseer las más variopintas formas. Normalmente estos dispositivos son leídos y restaurados cada cierto tiempo con el objetivo de controlar el nivel de radiación total (integrado) de cada persona y actual en caso de anomalías. Aquí mostramos algunos ejemplos:



Los rayos cósmicos

La siguiente fuente de radiación que pretendemos lograr detectar con nuestro proyecto son los rayos cósmicos. Los rayos cósmicos son conjuntos de partículas que viajan a velocidades muy parecidas a la de la luz y son por tanto **altamente energéticos** (varios TeV). No se sabe muy bien dónde se generan, pero sí sabemos que alcanzan nuestro planeta continuamente. Gracias a la atmósfera y al campo magnético terrestre, los cósmicos se descomponen en cascadas de partículas menos lesivas para el ser humano y que permiten su estudio *forénsico*. A estas descomposiciones se las conoce como cascadas atmosféricas o lluvia cósmica. No sólo se ignora qué tipo de partículas originan estas **cascadas**, sino que los científicos desconocemos su lugar de origen y el mecanismo capaz de impartirles semejantes velocidades y energías.



Por ejemplo, en la figura adjunta, un protón de 10^{15} eV penetra en la atmósfera a 35 km de altitud. Al nivel del suelo llegan unas 10^{16} partículas, de las cuales el 18% son gammas, otro 18% electrones y positrones, 1.7% muones y 0.3% hadrones.

El observatorio Pierre Auger, en Argentina, tiene como la finalidad la detección de estos rayos cósmicos y sus cascadas y es quizás el mejor laboratorio a nivel mundial dedicado a este fenómeno tan relevante. Nuestro desarrollo pretende dotar a cualquier persona interesada de un marco técnico e informático para la detección y registro de estas cascadas basado en los mismos principios utilizados en el Pierre Auger.

Las cámaras digitales del consumo

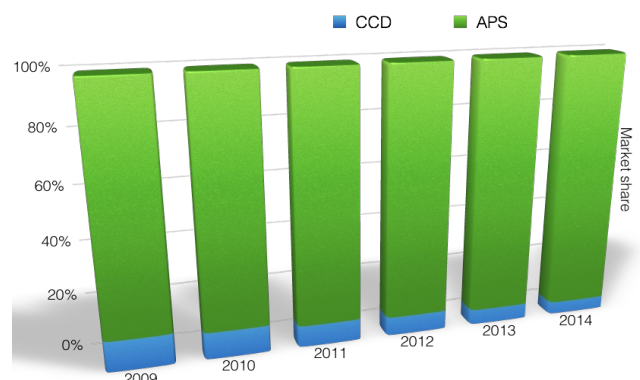
Las cámaras digitales son dispositivos ópticos y electrónicos que permite capturar imágenes sin necesidad del tradicional rollo de película ni proceso de revelado químico alguno. En la actualidad están muy extendidas y se usan de manera masiva. La adopción generalizada por parte de fabricantes de teléfonos móviles y ordenadores personales ha contribuido también a su universalización.

Su resolución (número de píxeles en fila por el número de píxeles en columna) varía mucho de fabricante a fabricante. Además de esta característica, podemos encontrar otras más relevantes y críticas para el correcto funcionamiento de nuestro experimento. Vamos a centrarnos en comentar la tecnología subyacente al sensor de luz que utilizan este tipo de cámaras. Podemos encontrar de dos tipos y ambos se basan, como no podía ser de otra manera, en el **efecto fotoeléctrico**.

Los **CCD** ó *Dispositivos de Carga Acoplada* contiene un número determinado de condensadores enlazados o acoplados. Bajo el control de un circuito interno, cada condensador puede transferir su carga eléctrica a uno o a varios de los condensadores situados en un circuito impreso cercano. La señal se produce dentro de un volumen de silicio cuando la luz en la zona polarizada genera electrones (carga) que se almacenan temporalmente en una *trampa*. Esta carga o señal eléctrica producida en cada pixel es enviada posteriormente al *exterior* y desde allí es amplificada por parte de electrónica adyacente.

Los **APS** ó *Active Pixel Sensor* son sensores que detectan luz basándose en tecnología CMOS. Los ventaja de esta tecnología es que integra más funciones en el mismo chip sensor, como por ejemplo controles de luminosidad o correctores de contraste. Cada pixel incorpora un amplificador de la señal eléctrica y es muy común encontrar el propio conversor analógico-digital en el mismo chip. La principal diferencia con un CCD es que cuando una partícula pasa por un pixel APS, no se genera (ni queda atrapada) ningún tipo de carga que podamos extraer y analizar. Las cámaras APS son más rápidas a la hora de capturar imágenes, si bien la calidad de las mismas es conocidamente inferior a la de las obtenidas con equipos fotográficos basados en CCD.

Por esta razón que acabamos de mencionar, **nuestro dispositivo experimental necesita de cámaras digitales de tipo CCD**. Si bien es verdad que su presencia en el mercado de los dispositivos



de captura digital se ha visto reducido en los últimos años –como puede verse en la gráfica adjunta–, su presencia es aún patente en cámaras digitales e incluso webcams de alta gama. Por ejemplo, la conocida cámara externa iSight de Apple estaba inicialmente basada en tecnología CCD. A medida que este conocido fabricante pasó a integrar las iSight en las propias pantallas de sus ordenadores, reemplazó también los sensores CCD por tecnología APS, sensiblemente más barata y adecuada para productos de consumo.

Al igual que la iSight externa (interfaz IEEE1394/FireWire), multitud de webcams de otros fabricantes a la venta entre los años 2000 – 2012, están basadas en APS.



Como hemos comentado, es muy normal encontrar sensores de tipo **CCD en cámaras fotográficas de gama media-alta** y que encajan perfectamente en el entorno experimental de nuestro proyecto. Aquí tenemos algunos ejemplos de las mismas:



Desarrollo del framework básico para clientes de escritorio y móviles

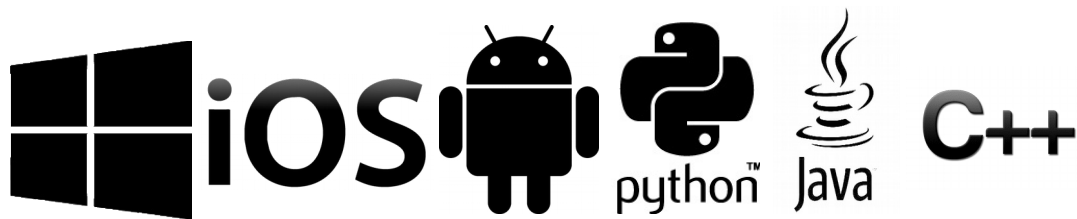
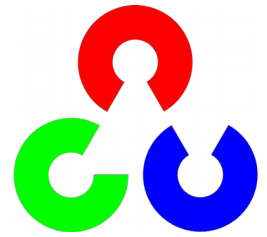
En esta sección pasamos a describir el esquema de desarrollo que hemos seguido para asegurar que el resultado de nuestro proyecto permanezca operativo a lo largo del **mayor número de plataformas informáticas y cámaras digitales** usadas hoy en día.

API para comunicación con cámaras integradas y externas

En nuestro proyecto necesitamos comunicarnos con el mayor número de cámaras digitales posibles y en el mayor número de entornos informáticos posibles. Podríamos delegar en tecnologías y APIs específicas de cada plataforma y cámara (como Video4Linux, Windows Media, QuickTime, etc.), pero añadiríamos muchas complejidades inesperadas e incompatibilidades. Si bien es verdad que existe un estándar de video sobre USB por parte del *USB Implementers Forum*, es muy corriente encontrar dispositivos de captura de imagen que aún no hayan suscrito dichas especificaciones. Lo mismo ocurre en el terreno del video sobre IP: a pesar del esfuerzo por lograr una estandarización, cada fabricante maneja su propio protocolo y sistema de funcionamiento. Por último, aunque en menor cantidad, existen también cámaras con interfaz FireWire. Justamente resulta interesante no descartar el soporte de este último tipo, pues suele tratarse de cámaras de video de gama media-alta y equipadas con sensores CCD de alta calidad, idóneos para nuestro proyecto.



Afortunadamente hemos podido contar la ayuda de la biblioteca de análisis de imagen y visión por computador **OpenCV** (opencv.org) para lidiar con este problema de **fragmentación tecnológica**. Se trata de un *toolkit* originalmente desarrollado por Intel. Se ha utiliza en infinidad de aplicaciones: desde sistemas de seguridad, detección de movimiento, control de procesos y reconocimiento de formas y objetos. Uno de los puntos fuertes de OpenCV es el **soporte masivo de cámaras de múltiples fabricantes**, a través de los más variados interfaces (USB, FireWire, serie, TCP/IP, etc.) con lo que no resulta muy atrevido asegurar que nuestro desarrollo cuenta con las mejores cartas para calificarlo de compatible con casi todas las cámaras digitales del mercado. Por otro lado, e igual de importante, es que OpenCV incorpora de serie interfaces de explotación para la gran mayoría de plataformas informáticas actuales y lenguajes de programación: iOS, Mac OS X, Windows, Windows Phone, Android, Java, Python y C++.



Herramientas para el análisis de imágenes

Una vez obtenidas las imágenes de las cámaras digitales, es necesario poder llegar a interrogar cada uno de los **pixeles** que las componen, y hacerlo de manera eficiente e **independiente de plataforma**. Nuevamente, la tentación inmediata es usar las APIs específicas de cada plataforma para este cometido: Quartz/NSImage en el caso de Mac OS X, GDI/Direct2D en el caso de Windows, por citar algunos ejemplos. Sin embargo, hemos optado por **Cimg**, una robusta e incipiente librería escrita en una única y elegante cabecera de C++ con las siguientes atractivas ventajas:



- La cabecera Cimg.h contiene todas las clases y funciones que componen la propia librería.
- No es necesario ningún tipo de compilado de la librería como tal. Esta se compila al mismo tiempo que nuestro código.
- No se dan dependencias que haya que satisfacer. Tan sólo con el citado fichero de cabecera, estamos listos para procesar cualquier tipo de imagen.
- Lo más importante. Cimg no realiza ningún tipo de compresión, pre- ó postprocesado y nos permite trabajar cómodamente *frame a frame*, **pixel a pixel** con la imagen original.

A pesar de estar escrita en C++, Cimg puede funcionar también en el sistema operativo Android si hacemos uso del *Native Development Kit* (NDK) que permite el desarrollo conjunto con el SDK de Android (este último basado en Java).

Uso del framework para detectar radiación ambiental

A continuación describimos las fases de uso del framework, una vez compilado y operativo. Inicialmente, como no podía ser de otra manera, conectamos nuestra cámara y la **protegemos todo lo posible de la luz ambiental**, que no haría otra cosa sino distorsionar el proceso de medida.



Proceso general de captura

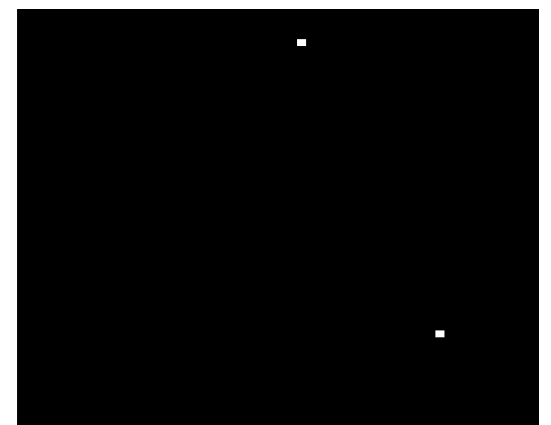
El análisis consiste en la adquisición de un número seleccionado de imágenes y su procesamiento de forma individual. En contra de lo que sería lo más lógico no estamos convirtiendo la imagen a escala de grises, sino que asumimos que los píxeles rojos, verdes y azules están juntos. El motivo tiene que ver con cómo se produce la señal en RGB, donde unos filtros de colores se anteponen al sensor para que cada elemento reciba una luz proporcional al color. En caso de que la carga del sensor no la genere luz si no una partícula ionizante, todos los píxeles funcionan de la misma forma.

En nuestro caso, se toman 25 imágenes por segundo y por tanto, 100 imágenes tardan unos 4 segundos en adquirirse. Por simplicidad, no son evaluados los tiempos muertos.

Calibración

Lo primera tarea que tenemos que hacer, una vez que ya podemos comunicarnos con nuestra cámara y comienzan a llegar imágenes, es calibrar el dispositivo. Todo dispositivo de medida necesita de una calibración. En nuestro caso consistiría en identificar los **umbrales** para los que estamos seguros que tomamos medidas reales (fruto del paso de radiación ambiental y cósmica por el sensor).

Inicialmente nos interesa conocer el **ruido de cada pixel** y el valor medio de señal que tiene. Para ello acumulamos el valor de cada pixel en cada imagen tomando para ello unas 800 imágenes. Posteriormente se realiza un rechazo que consistiría, en una primera aproximación, en seleccionar los píxeles con niveles de ruido más elevado. La siguiente imagen muestra el ruido de cada uno de los canales:



Otra parte también importante del proceso de calibración es la identificación de **píxeles muertos** o defectuosos, que hemos de tener en cuenta a la hora del funcionamiento operativo. En nuestro caso, por ejemplo, hemos identificado dos islas de píxeles con funcionamiento defectuoso o nulo. Pueden verse estos dos grupúsculos en la siguiente figura:

Una vez que tenemos los niveles de ruido identificado y los canales/píxeles dañados, construimos la **máscara de calibración** (pedestales) a partir de la cual ya podemos comenzar a tomar medidas. A continuación mostramos la máscara de pedestales que hemos obtenido con nuestra propia webcam:



Detección

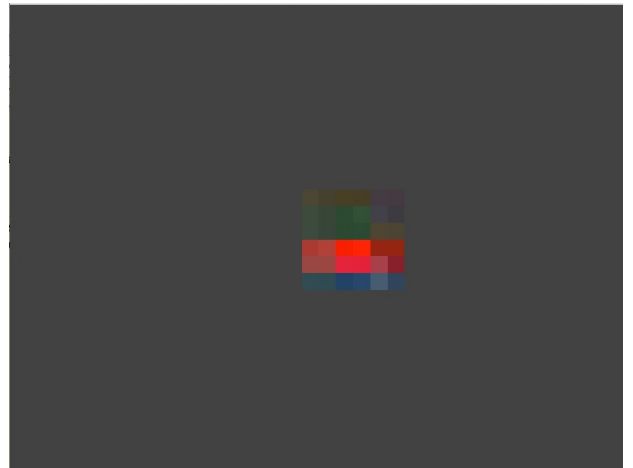
El análisis que proponemos consiste en la adquisición de un número seleccionado de imágenes, y su **procesado de forma individual**.

Cada imagen se subtrae y se compara con el nivel de ruido, si un pixel sobrepasa dicho umbral, se genera una **máscara binaria** (es decir construye una segunda imagen donde sólo los pixeles con el nivel relativo de señal/ruido ha sobrepasado el umbral. Estos se marca con un 1).

Esta máscara binaria suele contener pixeles adyacentes que están cerca del umbral y que contienen, por tanto, señal. Si se trata de una radiación ionizante la que está llegando al sensor, los pixeles contiguos pueden contener algo de carga pero no sobrepasar el umbral impuesto. Se realiza por tanto un **ensanchamiento de la máscara binaria**.

Este ensanchamiento además se etiqueta, o lo que es lo mismo en vez de asignar 1 a cada *mancha*, si se dan varias de estas de manera contigua, asignamos valores 1, 2 ó 3 que nos permite identificar cuántos *spots* tenemos en el detector. Una selección de pixeles de un spot en concreto se puede ver en la image adyacente.

Estas operaciones de análisis y **clustering** se realizan cómodamente y de manera multiplataforma gracias a la biblioteca Cimg que hemos descrito previamente.



Funcionamiento en modo detector *Geiger*

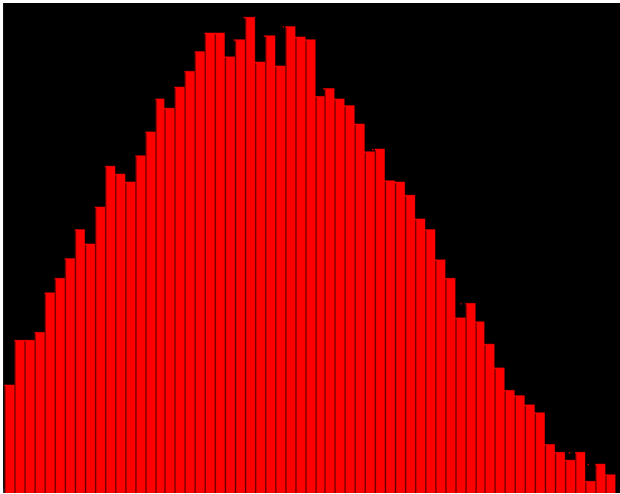
Una vez que ya tenemos la cámara calibrada y estamos detectando eventos relacionados con la radiación, podemos utilizar nuestra herramienta para operar en modo detector *Geiger-Müller* o cámara proporcional. En este modo podemos establecer dos frecuencias: la que tenemos cada 100 adquisiciones (*frecuencia instantánea*) y la frecuencia promediada. **Si la comparación entre frecuencia instantánea y promediada cambia, entonces estamos acercándonos a ubicaciones donde pueden existir fuentes radioactivas.** Aquí mostramos algunas frecuencias observadas con nuestro *setup* específico.



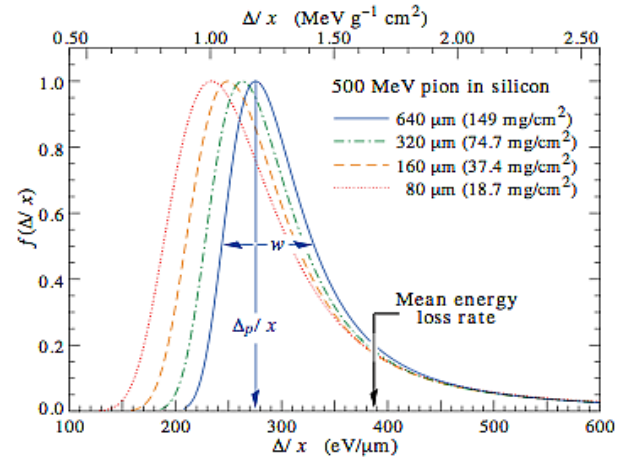
0.0508571 Hz 0.0490667 Hz 0.048 Hz 0.0470588 Hz 0.0462222 Hz 0.0450526 Hz

Funcionamiento como sensor de trazas

En este modo la cámara captura de manera continua posibles eventos. Los eventos son clasificados según la energía relativa depositada y es posible recrear un histograma con estas medidas. En el eje y están representados el números de eventos positivos y en el eje x, cada bin se corresponde con un valor entre 0 y 255 en intensidad o nivel de gris. A modo de ejemplo, la siguiente imagen muestra las medidas obtenidas durante 48 horas de actividad. Los bines han sido agrupados de 8 en 8.



27. Passage of particles through matter



Esta curva corresponde muy bien a la esperada en el **paso de una partícula de rayos cósmicos por la materia** (denominada *curva de Lambdau*). En este caso las largas colas se ven reducidas por dos motivos:

1. El rango dinámico (la respuesta del detector CCD) está limitado a 8 bits
2. Parte de la carga se reparte entre los distintos campos de la imagen (píxeles pares e impares) y los distintos colores (RGB) y el algoritmo se ha simplificado para tratar el sensor como como tres sensores.

Conclusiones y entorno colaborativo: [rad@home](http://lhcphe07.ific.uv.es/rad_at_home/detectorccd.tgz)

Se ha presentado una forma de obtener variaciones de radiación mediante el empleo de **código sencillo**, manipulable y que usa **librerías** de adquisición y tratamiento bastante **populares**. El código fuente de la primera revisión está en http://lhcphe07.ific.uv.es/rad_at_home/detectorccd.tgz. Con él no sólo se puede aprender cómo **detectar la radiación** sino cómo funciona todo el **mecanismo de captura de imágenes** y métodos sencillos de cálculo para extraer información.

Una vez que ya podemos realizar medidas de radiación con nuestra propia cámara y entorno informático, proponemos:

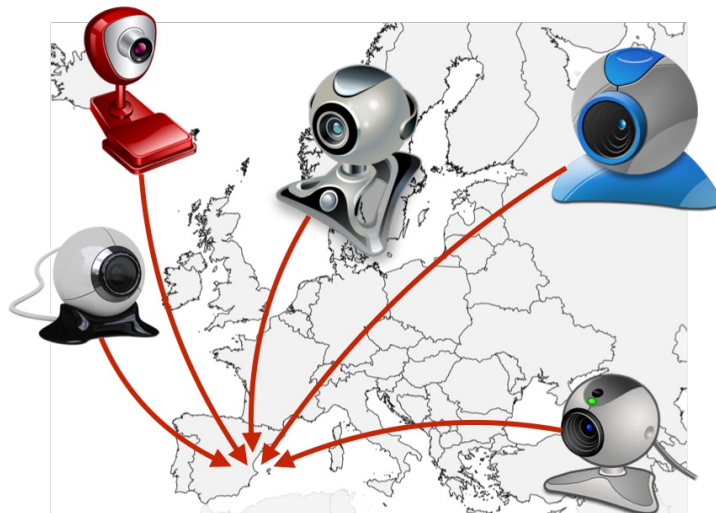
- Comparar la radiación entre entornos abiertos de entornos cerrados (10 minutos de adquisición). Esta medida puede dar una idea de la cantidad de materiales que emiten radón. Se realiza calculando la frecuencia del sensor dentro de una habitación cerrada y en un espacio abierto.
- Calcular la dosis de reflejo de una radiografía. Es decir, medir durante una exposición radiográfica (centrada en la zona de interés) con una cámara o móvil la radiación que los colimadores de los dispositivos radiográficos no nos envían directamente.
- Contar muones de forma comparativa. En modo sensor de trazas comparar los muones que detecta el sensor a nivel de tierra o cuando se viaja en avión (lo mismo en *modo geiger*).

Podemos escribir la aplicación que queramos para nuestro **detector de radiación portable casero**.

Queremos recalcar la **importancia** a nivel divulgativo y educativo de la promoción de **herramientas de software libres**, estándar y **multiplataforma**, también su uso en **ámbitos científicos**. Deseamos resaltar la trascendencia, a nivel educativo, de saber manejar y gestionar distintos *toolkits*, librerías y herramientas de programación y cómo estos pueden orientarse hacia la producción científica, y, en concreto, el estudio de la física básica.

Por último nos gustaría enfatizar el increíble potencial que tienen este tipo de escenarios de investigación personal e individualizada. Los resultados de uso de estas herramientas, aunque inicialmente tienen un carácter aislado, pueden acabar formando parte de **entornos colaborativos y sociales**. Por ello, aunque no lo hemos planteado directamente como fruto de este proyecto de investigación, dejamos la puerta abierta a la implantación de un sistema centralizado y social donde cada usuario del framework que en este texto hemos introducido, pueda enviar de manera automatizada los resultados de medidas que haya llevado a cabo.

El objetivo no sería otro que construir una **red de estudiantes y personas aficionadas a los detectores** y a la física en general. Esta sistema o red, sería el encargado de centralizar mediciones caseras de radiación ambiental y ponerlas a disposición de la comunidad educativa e incluso científica, ayudando a complementar el noble papel iniciado por proyectos como el observatorio Pierre Auger, que hemos comentado en la introducción.



Referencias

- http://lhcphe07.ific.uv.es/rad_at_home/
- http://lhcphe07.ific.uv.es/rad_at_home/detectorccd.tgz
- http://es.wikipedia.org/wiki/Radiación_cósmica
- http://www.unscear.org/docs/reports/2008/09-86753_Report_2008_Annex_B.pdf
- http://en.wikipedia.org/wiki/Charge-coupled_device
- <http://cimg.sourceforge.net>
- <http://opencv.org>
- CCDs vs APS <http://www.digitalbolex.com/global-shutter/>
- <http://pdg.lbl.gov/2005/reviews/passagerpp.pdf>