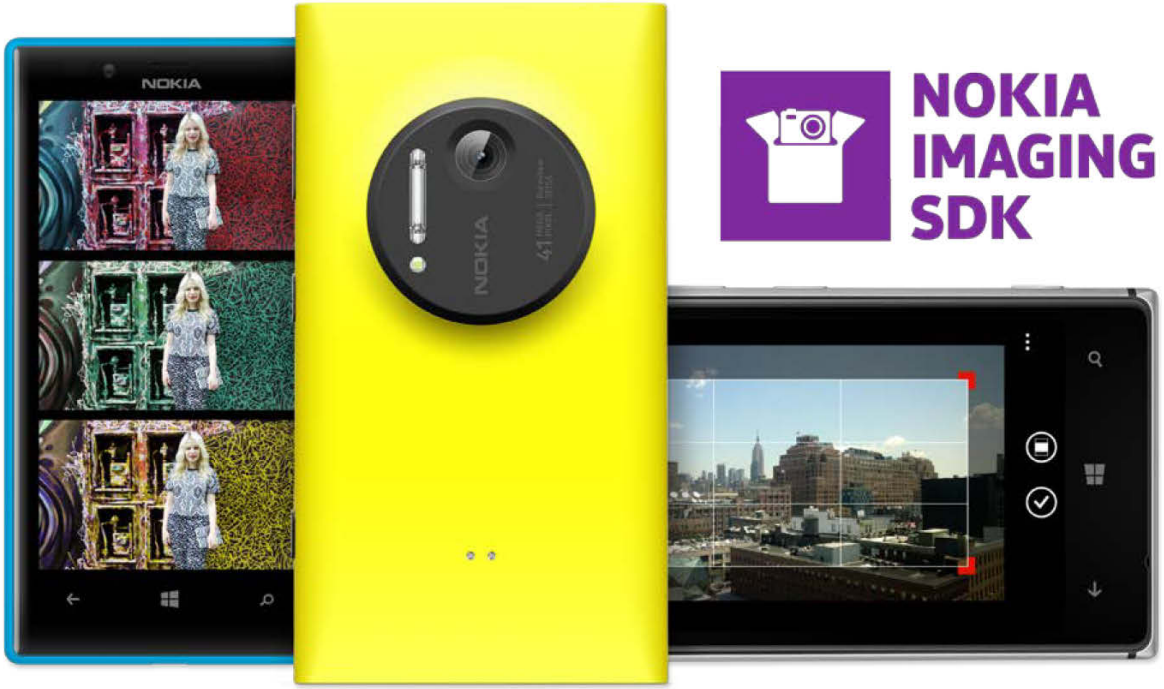
LUMIA APP LABS #15

USING THE NOKIA IMAGING SDK



Berthier Lemieux
Technology Wizard

NOKIA Developer

SCALADO

Experts in imaging algorithms

Acquired by Nokia, 24 July 2012

Their technology drives:

Nokia Smart Camera

Nokia Cinemagraph

Nokia Creative studio

NOKIA IMAGING SDK

Really fast preview of any region of the image

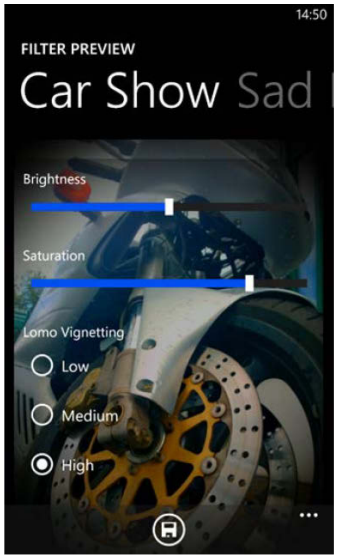
Complete set of base filters and effects (52 filters).
Combine them to generate more complex ones

Highest performance and low memory consumption through RAJPEG technology (>15 patents)

Cropping and JPEG compression parameter control

Example projects

Filter Effects, simple illustration of basics



Real-time Filter Demo, live viewfinder stream processing

Filter Explorer, complex illustration of various usage scenarios

GOOD TO KNOW

- Apps developed with the SDK can be used on **any Windows Phone 8**, including non-Nokia, devices
- The library is a WinPRT library. Routines are available via a **C# interface as well as a C/C++ interface**
- Nokia Imaging SDK License Agreement: **Free to use, attribution**

List of filters and effects I

Antique	Auto Enhance	Auto Levels	Blend	Blur	Brightness
					
Cartoon	Color Adjust	Color Boost	Colorization	Color Swap	Contrast
					
Crop	Curves	Despeckle	Emboss	Exposure	Flip
					

List of filters and effects II

Fog	Foundation	Frame	Free Rotation	Grayscale	Grayscale Negative
					
Hue Saturation	Image Fusion	Levels	Local Boost	Lomo	Magic Pen
					
Milky	Mirror	Mono Color	Moonlight	Negative	Noise
					

List of filters and effects III

Oily 	Paint 	Posterize 	Sepia 	Sharpness 	Sketch 
Solarize 	Split Tone 	Spotlight 	Stamp 	Step Rotation 	Temperature and Tint 
er Vignetting 	Warp 	Watercolor 	White Balance 		

General overview of required steps

- Step 1: Include Nokia Imaging SDK Libraries into your project
- Step 2 : Create EditingSession
- Step 3 : Create and add filters to EditingSession
- Step 4 : Use asynchronous methods RenderToImageAsync or RenderToJpegAsync to produce the final processed image

Step 1: Nokia Imaging SDK Libraries

- Use NuGet to install the SDK libraries.
- Finalise the installation by :
  - Making sure that in the Project's Configuration Manager, only the **X86** and **ARM** platforms are listed. The **Any CPU** platform should be removed
  - Save your project in Visual Studio
  - With an external editor (notepad, for example), edit the .csproj file to support both X86 and ARM platforms. In the <HintPath> elements, replace "X86" by "$(Platform)".

PROCESSOR ARCHITECTURES

Target	Processor architecture type
Emulator	X86
Phone	ARM

You don't need to know/understand this to use the SDK!
Consider this as FYI.

	Language	Compiled to	Runs in a virtual machine ?	Processor architecture dependant?
Managed code	C#, VB	Intermediate Language (MSIL)	Yes (CLR)	No
Native code	C++	Machine language	No	Yes

Step 2: Create EditingSession

- Create an Imaging SDK EditingSession using a compressed or uncompressed image:
  - From a Stream (from PhotoChooserTask):

```
EditingSession session = await CreateEditingSessionAsync(stream);
```

  - From a JPEG in a IBuffer:

```
EditingSession session = new EditingSession(jpegData);
```

- From a WriteableBitmap:

```
EditingSession session = new EditingSession(sourceBitmap);
```

Step 3: Create and add filters and effects

- Use FilterFactory to create filters and effects
- Use EditingSession methods to add filters and effects:

```
session.AddFilter(FilterFactory.CreateCartoonFilter(true));
session.AddFilter(FilterFactory.CreateFogFilter());
```

- You can also use FilterGroup to add several filters and effects in one call

Step 4: Produce final processed image

You can render the processed image to:

- A XAML Image control:

```
await session.RenderToImageAsync(FilteredImage);
```

- A WriteableBitmap :

```
await session.RenderToWriteableBitmapAsync(FilteredBitmap);
```

- An IBuffer :

```
IBuffer jpegOut = await session.RenderToJpegAsync();
```

RESOURCES

You found an error in the SDK, have suggestions, need help?

Nokia Imaging discussion board: http://nokia.ly/DiBoImg

You have developped an app with the SDK?

We'd love to hear about it. Tell us by sending a mail at developer-relations.marketing@nokia.com

Documentation and code samples

- Imaging in the Lumia Developer's Library: http://nokia.ly/WP_lib_img
- Nokia Imaging SDK: http://www.developer.nokia.com/imaging

THANK YOU!

NOKIA IMAGING WIKI COMPETITION 2013Q3

Write a great tutorial, guide or an article with code which shows how to use the Nokia Imaging SDK or Camera and any other Windows Phone APIs related to imaging in useful, imaginative and innovative ways and/or provide the best feedback on the Nokia Imaging SDK.

developer.nokia.com/Community/Wiki/

NOKIA Developer