Technologies to Reduce the Access Barrier **TrabHCI** in Human Computer Interaction Erasmus Intensive Programme IP29588-1-1731-10

C#, WPF and the .NET Framework

2

Michela Goffredo, Ivan Bernabucci University Roma TRE goffredo@uniroma3.it



C# is a general-purpose, object-oriented programming language.

C# includes encapsulation, inheritance, and polymorphism.

C# has some features in his structure:

- Unified type system: all types derive from a base type
- There are different types: objects, interfaces, structures, enumerations (like Java) and delegates!
- Function members: methods, events, properties



C# derives, like Java, the main features of C++ simplifying several aspects:

- No pointers required
- Automatic memory management through Garbage Collection
- Use of collections (List, Queue, ...)
- Lambda expressions
- *dynamic* keyword



The .NET Framework is a software platform for building systems and applications.

It consists of the runtime **CLR** (Common Language Runtime) and several libraries.

This means that there is a common runtime engine that all the .NET languages share together, and that is possible to exploits components of other languages (F#, Visual Basic, Delphi.Net, IronPython.NET, J#)



There is a **Base Class Library** inside the .NET Framework that handles low-level operations such as:

- Database access
- File I/O
- Threading
- • •

Like the *Virtual Machine* in Java, the .NET Frameworks compilers provides an intermediate layer between the programming language and the assembly code: *Intermediate Language* (like the *bytecode*) which will be then used by the .NET framework in run-time execution.

This IL is the managed code (it can be .dll or .exe).



An example of IL:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
        static void Main(string[] args)
        {
            Calc c = new Calc();
            Console.WriteLine("3 + 5 is {0}",c.Add(3,5));
        }
    }
    class Calc
    {
        public int Add(int x, int y)
        {
            return (x + y);
    }
}
    6
```



If we use the ildasm.exe and we open the Add method of the calculator this is what we see -> non platform-specific instructions

.method public hidebysig instance int32 Add(int32 x, int32 y) cil managed

// Code size 9 (0x9)
.maxstack 2
.locals init ([0] int32 CS\$1\$0000)
IL_0000: nop
IL_0001: Idarg.1
IL_0001: Idarg.2
IL_0003: add
IL_0003: add
IL_0004: stloc.0
IL_0005: br.s IL_0007
IL_0007: Idloc.0
IL_0008: ret
} // end of method Calc::Add



Build Applications with Visual Studio 20XX

Even if it's possible to write in Notepad and compile with the prompt command csc.exe

(ex: csc /target:exe Car.cs)

The IDE Visual Studio offer some advantages:

- Support for visual design
- Intellisense
- Free templates for handling Xbox, WPF, Twitter..



Building an application:

Referencing external assemblies ('..yes, the Internet is full of object extensively tested which can be included in your project and will simplify your development..')

- 1. Right click in the References Folder of the Solution Explorer on the right side
- 2. Select Add Reference
- 3. Browse for your reference and click Add

Adding a new Class

- 1. Right click in the Project Icon of the Solution Explorer on the right side
- 2. Select Add Class



0	WpfApplication1 - Microsoft Visual Studio	
File Edit View Project Build Debug Team Data Tools Test Wind 「「・ ・ 」・ 」 」 』 よ い し ・ ・ ・ ・ ・ ・ ・ Debug 「 い い い … 」 … 「 年 年 三 2 ー の つ の の の 。 。	low Help ▼ 🖄 💎 ♥ 🔜 🖸 ▼ 🚽	
MainWindow.xaml × MainWindow.xaml.cs		Solution Explorer



0	WpfApplication1 - Microsoft Visual Studio		
File Edit View Project Build Debug Team Data Tools Test Window Help	- 🔍 🕾 🕼 🕸 🏷 🖬 🖬 🗆 - 🚽		
MainWindow.xaml × MainWindow.xaml.cs			Iution Explorer
	New Item Ctrl+Shift+A Existing Item Shift+Alt+A New Folder Window Page User Control Resource Dictionary Class Shift+Alt+C	Add Add Reference Add Service Reference View Class Diagram Set as StartUp Project Debug Add Solution to Source Control X Cut	► Tea Start Class Ctrl+X
<pre>Design</pre>	ation"	Paste Remove Rename Unload Project Open Folder in Windows Explorer Properties	Ctrl+V Del WpfApplication1.cs er c:\users\ivan\docu Alt+Enter
<pre> 100 % </pre> Image: The second			roject File he name of the file containing uild, configuration, and other inf



```
class Program
    {
        static void Main(string[] args)
        {
            MultiDimensionalArray();
            Console.ReadLine();
        }
        static void MultiDimensionalArray()
        {
            int[,] myMatrix;
            myMatrix = new int[6, 6];
            // Populate Array
            for (int i = 0; i < 6; i++)
                 for (int j = 0; j < 6; j++)</pre>
                     myMatrix[i, j] = i * j;
            // Print the Array
            for (int i = 0; i < 6; i++)</pre>
             {
                 for (int j = 0; j < 6; j++)</pre>
                     Console.Write(myMatrix[i, j] + "\t");
                 Console.WriteLine();
            }
        }
    12
```

Multidimensional Arrays



```
static void Main(string[] args)
                                           public struct Point
        {
            Point p;
                                                       public int X;
            p.X = 10;
                                                       public int Y;
            p.Y = 20;
                                                       public void Increase()
            p.Increase();
            p.Display();
                                                           X++; Y++;
            p.Decrease();
            p.Display();
                                                       public void Decrease()
            Console.ReadLine();
                                                           X--; Y--;
        }
                                                       public void Display()
                                                            Console.WriteLine("X = {0}; Y = {1}",X,Y);
```

Structure...like lightweight classes type!...

They are *value type*, not *reference type*

Point p2 = p they are not the same thing!!



C# Class:

Formally a class is composed by:

- Field data (the member variables)
- Members that operate on these data (constructor, properties, methods, events)

```
class ECG
{
    // Thes state of the object
    private string patientName;
    private int samplingFrequency;
    private List<double> dataSamples;

    public int MeanValue()
    {
        return (int)(dataSamples.Sum() / dataSamples.Count);
    }
}
```



```
class Program
    ł
        static void Main(string[] args)
        ť
            ECG myECG = new ECG();
            myECG.patientName = "Mario Rossi";
            myECG.samplingFrequency = 1000;
class ECG
    ł
        // The state of the object
        public string patientName;
        public int samplingFrequency;
        public List<double> dataSamples;
         // Methos of the object
        public int MeanValue()
            return (int)(dataSamples.Sum() / dataSamples.Count);
    }
```

15



The Windows Presentation Foundation is a graphical display system for Windows.

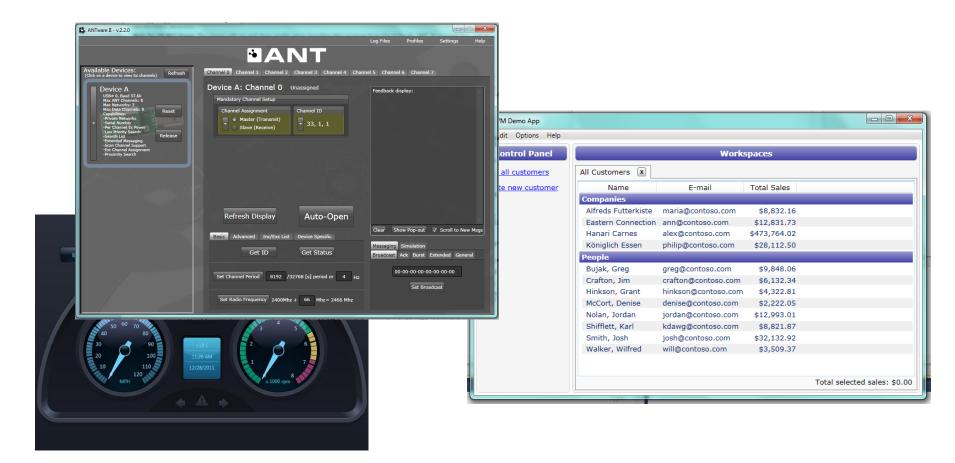
Windows left the GDI/GDI+ (used for more than 10 years) system to embrace the DirectX libraries (best performance)

- WPF enables automatically video card optimization
- and when the video card is too old,..
- ..it automatically optimizes the software (DirectX functions)



🖳 Form1				- • •
	C. Data	······		Q
Configure Master		Left Pedal	Right Pedal	
Start Simulation	Efficency Index			
Stop Simulation	Total Power			
	o Power			Q
	Speed (rps)	Q <u> </u>		
•	0	00		







http://archive.msdn.microsoft.com/wpfsamples

WPF allows the design of stylish and high-performant application (the programmer should work with a real designer!!):

- Web Layout Model (flexibility)
- Rich Drawing Model (transparent, shapes, graphical layers)
- Animation and timeline
- Support for Audio and Video (Windows Media Player)
- Styles and Template



WPF is based on XAML (Extensible Application Markup Language - 2009)

Usually XAML is not written by hand but graphically design by means of special tools (like Expression Blend or Visual Studio design section)

The idea under the XAML is to separate completely the graphic part from the coding part



🗙 Start Page - Microsoft Visual Studio	Production of the last	4	1 11 61 Dr 61	4 4 4	2. 0	4.4		x
File Edit View Debug Team Data Tools Test Window								
		- 🦉 _spaceForXAxis	- 🔍 🕾 🖬 🕺 🏷 🛃 💷 - 🖕					
Start Page X						 Solution Explorer 	▼ 9	×
Tool								
Visual Studio [®] 2010 Professio								
VISUAI SLUCIO 2010 Professio	onal							
Connect To Team Foundation Server	Get Started Guidance a	and Resources Latest News						
	Welcome Windows Web	O Cloud Office SharePoint Data						
C New Project								
Open Project	Reveal Analysis and Analysis (and Analysis and Analysis) (and Anal	What's New in Visual Studio 2010						
	CONTRACTOR AND	Learn about the new features included in this r	elease.					
Recent Projects	Nager 1007 1	Visual Studio 2010 Overview						
		What's New in .NET Framework 4 What's New in Visual C#						
EMGUTest	A CANANA AND AND AND AND AND AND AND AND AN	Customize the Visual C#						
BaveLoadKinectFrames	Annual and an an							
Emgu.CV.Example								
ProgettoPostura2	Creating Appl	lications with Visual Studio						
ProgettoPostura2 KinectWPFDemo -								
TestOldPedals								
Emgu.CV	Extending Visu	ual Studio						
WpfApplication7	E and a second s							
Microsoft.Samples.Kinect.WpfViewers	177 (Children and Children and							
Microsoft.Samples.Kinect.WpfViewers								
Microsoft.Samples.Kinect.WpfViewers	Community an	nd Learning Resources						
SettingUpDevEnvironment								
KinectSDKSamples(C#)								
KinectExplorer								
 Close page after project load Show page on startup 								
Error List					- ₽ :	×		
3 0 Errors 0 Warnings 0 Messages								
Description			File	Line Column	Project			
						-		
						Representation Explore	r 📷 Team Explorer 🔌 Class View	
Ready								
🧑 🖸 餐 🖉 🧔		🤣 💐			1.1	П 🔺	to 👽 🔁 🛱 🗘 12:28 PM 3/22/201	



licrosoft Visual Studio									_ 0
Edit View Debug Team Data Tools Test Wi	indow Help		-	Trans Annual States	and the state of the				
	B Debug - A	ny CPU	- 🖄 _spa	reForXAxis	- 🗟 🖀 📑 🖄 🏷 🛃 🖳 🗆				
		.,				•			
								Solution Explorer	*
	New Project						?	×	
	Recent Templates		.NET Framework 4	 Sort by: Default 		Search Installe	d Templates	Q	
	Installed Templates		_			Type: Visual			
	✓ Visual C#		C# Windows	Forms Application	Visual C#		sentation Foundation clie		
	Windows					application	sentation Foundation clie	ent	
	Web		WPF App	lication	Visual C#				
	▷ Office Cloud		Console /	Application	Visual C#				
	In the Box		Console						
	Reporting		Class Libr	ary	Visual C#				
	SharePoint								
	Silverlight Test		C# WPF Brow	vser Application	Visual C#				
	WCF		C# Empty Pr		Visual C#				
	Workflow		C# Empty Pr	ojeci	Visual C#				
	Other Languages		c ≇ Windows	Service	Visual C#				
	 Other Project Typ Database 	es							
	 Test Projects 			tom Control Library	Visual C#				
	Online Templates								
	onine rempiates		WPF User	Control Library	Visual C#				
			c ≇ Windows	Forms Control Library	Visual C#				
				ronno contror cionary	risdar es				
	Name:	TestApp							
				- 2010) Durit-	•	Desure			
	Location:		locuments\visual studi	o 2010\Projects	•	Browse			
	Solution name:	TestApp				Create directo Add to source			
Fror List						state to source		×	
3 0 Errors 1 0 Warnings 1 0 Messages							OK Cance		
Description					File	Line	Column Project		
,									
								Solution Explorer	Team Explorer 🛛 💐 Class Vie



• TestApp - Microsoft Visual Studio		and the second se	Course and the second s			-		_ 0 <u>_ x</u>
File Edit View Project Build Debug Team Data	Tools Test Window Help							
Si N> N> M≥ K2 IS IIS III II								
🛐 • 📴 • 💕 🛃 🥔 👗 🛍 🛍 👘 • ભ • 💭 • 🗈	🖳 🕨 Debug 🔹 x86	- 🧭 _spaceForXAxis	- 🔍 🕾 📑 🐼 🏷 🛃 🖳 -	_				
〒 ▲ ▲ 惛 建建 三 일 □ ₽ ♀ ぬ は								
MainWindow.xaml × MainWindow.xaml.cs ×							Solution Explorer	- 4 ×
c:\users\ivan\doc	cuments\visual studio 2010\Projects\TestA	pp\TestApp\MainWindow.xaml.cs				/ n	Solution 'TestApp' (1	project)
	MainWindow						Image: A star and a star a	
							 Properties References 	
							App.xaml	
×							MainWindow.	xaml
						E		
	U					*	1	
□ Design ↑↓ □ XAML 1 □ <window <="" td="" x:class="TestApp.MainWindo</td><td>ow"><td></td><td></td><td></td><td></td><td></td><td></td><td></td></window>								
2 xmlns="http://schemas.micr	rosoft.com/winfx/2006/xaml/preser	ntation"				*		Team Explorer 🛛 🕰 Class View
3 xmlns:x="http://schemas.mi 4 Title="MainWindow" Height=	icrosoft.com/winfx/2006/xaml" ="350" Width="525">							
5 🖻 <grid></grid>	- 550 WIGCH- 525 V					E	Properties	~ ↓ ×
6 7							TestApp Project Propertie	s •
8							811 2↓ 🖻	
9							 Misc Project File 	TestApp.csproj
100 % - 1						*	Project Folder	c:\users\ivan\documents\visua
🔄 Window Window 🕨								
Error List						- ₽ ×	1	
3 0 Errors 1 0 Warnings 0 Messages							1	
Description			File	Line (Column P	roject	1	
							1	
							Misc	
leady								



The XAML code behind the default form:

```
</Window>
```

- The element in a XAML maps to instance of .NET classes. The name of the element matches the name of the class (<Grid> is a Grid Object)
- You can nest elements inside elements (same way an HTML page is structured)
- Properties are set through attributes



Let's modify:

```
<Window x:Class="TestApp.MainWindow"</pre>
         xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation
         xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="200"></ColumnDefinition>
            <ColumnDefinition Width="*"></ColumnDefinition>
        </Grid.ColumnDefinitions>
         <Grid.Background>
            <LinearGradientBrush>
                <LinearGradientBrush.GradientStops>
                    <GradientStop Offset="0.00" Color="Red" />
                    <GradientStop Offset="1.00" Color="Violet" />
                </LinearGradientBrush.GradientStops>
            </LinearGradientBrush>
        </Grid.Background>
```

</Grid>
</Window>



estApp - Microsoft Visual Studio	and becaute the second	E Branch Huges							
Edit View Project Build Debug Team Data Te	ools Test Window Help								
) °⊳ °⊳ № № □, □, □, □ □ □, -] • □ • 22 , 23 3, □, 10 •7 • (* • , □ • □,	Debug - x86	- MarceForXAxis	- 🔍 🕾 🎲 🕸 🔆 🛃 🖳						
, 영 / 영 / 일 / 일 / 이 입 / 이 신 영 영 영 / 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이				T					
									- 1
MainWindow.xaml × MainWindow.xaml.cs						-	Solution Explorer		• 4
100%						- -	Solution 'TestApp' (1		
	MainWindow	281*					⊿ 📴 TestApp		
							 Properties References 		
							References		
*							MainWindow.	xaml	
						E			
			6						
□ Design ↑↓							1		
1 ⊡ <window http:="" schemas.micro<br="" x:class="TestApp.MainWindow</td><td></td><td></td><td></td><td></td><td></td><td>4</td><td></td><td></td><td></td></tr><tr><td>2 xmlns=">3 xmlns:x="http://schemas.mic</window>	<pre>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>	ntation"				-	Solution Explorer	Team Explorer 🛛 🗟 Cla	lass View
4 Title="MainWindow" Height="						E	Properties		– ą
5 S <grid></grid>						-	Grid <no name=""></no>		
7 ColumnDefinition Width							Properties 🖋 Even		
8 <columndefinition width<br="">9 </columndefinition>	="281*" />							15	
10 dia Contractional Contraction (Contraction Contraction Contract							Search		
100 % - 4						Þ	DataContext	Binding	
Grid Window/Grid							Style	Resource	
Error List						- 4 ×		 (Collection) 	
3 0 Errors 👔 0 Warnings 🚺 0 Messages							RowDefinitions	(Collection)	
Description			File	Line	Column	Project	⊿ Layout		
,						,	Width	🛛 Auto	
							Height	Auto	
							HorizontalAlignment	Stretch	
							VerticalAlignment	Stretch	
							Manaia		
							Margin MinWidth	0	

D



Let's modify:

```
<Window x:Class="TestApp.MainWindow"</pre>
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
   </Grid.Background>
        <Button Grid.Column="0" Content="Button1" Height="23"
HorizontalAlignment="Left" Margin="5,20,0,0" Name="button1" VerticalAlignment="Top"
Width="75" />
        <Button Grid.Column="0" Content="Button2" Height="23"
HorizontalAlignment="Center" Margin="0,60,0,0" Name="button2" VerticalAlignment="Top"
Width="75" />
        <Button Grid.Column="0" Content="Button3" Height="23"
HorizontalAlignment="Right" Margin="0,100,0,0" Name="button3" VerticalAlignment="Top"
Width="75" />
        <Image Grid.Column="1" Height="163" HorizontalAlignment="Left"</pre>
Margin="38,31,0,0" Name="image1" Stretch="Fill" VerticalAlignment="Top" Width="308"
/>
    </Grid>
</Window>
```



- Tashina - Minangh Wangi Chullin							
a TestApp - Microsoft Visual Studio File Edit View Project Build Debug Team Data Format Tools Test Window Help	Real programs include the						
	🎽 _spaceForXAxis 🔹 🖓 🚰 🚮	🖄 🏷 💽 🛃 🖂 🗸					
다 M							
MainWindow.xaml* × MainWindow.xaml.cs				•	Solution Explorer	_	- ₽ ×
					🔚 🍙 🛃 🗵 📰 🖧		
MainWindow					Solution TestApp (1 p) ⊿ I TestApp	roject)	
55°	4				Properties		
Button1					 References App.xaml 		
					App.xami	iml	
Button2					_		
Button3							
				E			
□ Design							
16 (/LinearGradientBrush>				\$			
<pre>17 18 <button content="Button1" grid.column="0" height="23" horizontalalign<="" pre=""></button></pre>	ment="Left" Margin="5,20,0,0" Name="button1	" VerticalAlignment	="Top" Width="75" /	*	🖏 Solution Explorer 🛛 📷 T	eam Explorer 🛛 🗟 Cla	ss View
19 <button content="Button2" grid.column="0" height="23" horizontalalign<="" p=""></button>	ment="Center" Margin="0,60,0,0" Name="butto	n2" VerticalAlignme	nt="Top" Width="75"	/>	Properties		- ₽ ×
20 <button content="Button3" grid.column="0" height="23" horizontalalign<br="">21 <image grid.column="1" height="163" horizontalalignment="Left" margin<="" p=""/></button>					Image image1		
22	· · · · ·			E	Properties 4 Events		
23 <pre></pre> 23 <pre>24</pre>						5	
				*	🔡 🏚 📮 Search		×
100 % 👻 🛛				•	✓ Common		*
Image (image1) Window/Grid/Image					DataContext	Binding	
Error List				▼ ₽ ×	Style	Resource	Ű
3 0 Errors 🛕 0 Warnings 🕕 0 Messages					Source	□	
Description		File	Line Column	Project	Stretch StretchDirection	 Fill Both 	
				-	▲ Layout	La both	
					Width	308	
					Height	163	
					HorizontalAlignment	Left	
					VerticalAlignment	🔶 Тор	
					Manufa	▲ 20 21 0 0	



Data binding is a relationship that tells WPF to extract some information from a source object and use it to set a property in a target object.

It's perfect for design decoupled systems. The View and the Logic.



First we talk about OpenCV!!

What is **OpenCV**?

OpenCV is an open source computer vision library (*http://SourceForge.net/projects/opencvlibrary*). The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, Matlab, and other languages.

It is highly-optimized for image processing -> Focus on real time applications



Open CV contains over 500 functions that span many areas in vision, including:

- Medical imaging
- Security
- User interface
- Camera calibration
- Stereo vision
- Robotics

A lot of applications have been released:

- Stitching images together in satellite and web maps
- Image scan alignment
- Medical image noise reduction
- Object analysis
- Security and intrusion detection systems
- Military applications



OpenCV can be used in commercial product without problem and its community counts more than 20.000 members..!!

Many time in Computer Vision there is the **transformation** of data from a still or video camera into either a **decision** (turning a color image into a grayscale image) or a new **representation** ("there are 5 tumor cells", "the person isn't part of the group")

While the brain has an internal auto-color setting, auto focus setting and pattern recognition system...

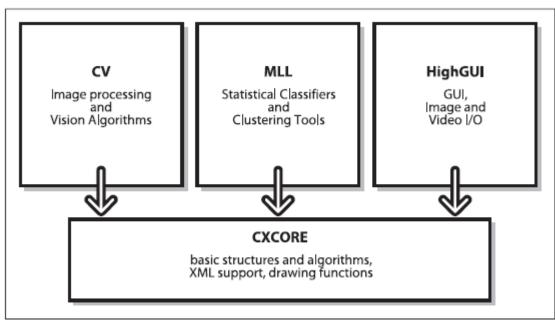
..This is what we get form a camera!!

194	210	201	era s	199	213	215	195	178	158	182	/209
180 114	189 126	190 140	221 188	209 176	205 165	191 152	167 140	147 170	115 106	129 78	163 88
87 102	103 112	115 106	154 131	143 122	142 138	149 152	153 147	173 128	101 84	57 58	57 66
94	95	79	104	105	124	129	113	107	87	69	67
68 41	71 56	69 68	98 99	89 63	92 45	98 60	95 82	89 58	88 76	76 74	67 65
20	41	69	75	56	41	51	73	55	70	63	44
50 72	50 59	57 53	69 66	75 84	75 92	73 84	74 74	53 57	68 72	59 63	44 37 42
67	61	58	65	75	78	76	73	59	75	69	50



OpenCV is aimed at providing the basic tools needed to solve computer vision problems.

In some cases, high-level functionalities in the library will be sufficient to solve the more complex problems in computer vision. Even when this is not the case, the basic components in the library are complete enough to enable creation of a complete solution of your own to almost any computer vision problem.





Basic types of OpenCV (they are all simple structures):

- CvPoint
- CvSize
- CvRect

The most important class in openCV is the **lpllmage**!!

It derives from the class CvMatrix (everything in OpenCV is a matrix), and this is the reason why it's possible to operate with special matrix functions and operators directly on these images!!



Function	Description
cvAbs	Absolute value of all elements in an array
cvAbsDiff	Absolute value of differences between two arrays
cvAbsDiffS	Absolute value of difference between an array and a scalar
cvAdd	Elementwise addition of two arrays
cvAddS	Elementwise addition of an array and a scalar
cvAddWeighted	Elementwise weighted addition of two arrays (alpha blending)
cvAvg	Average value of all elements in an array
cvAvgSdv	Absolute value and standard deviation of all elements in an array
cvCalcCovarMatrix	Compute covariance of a set of <i>n</i> -dimensional vectors
cvCmp	Apply selected comparison operator to all elements in two arrays
cvCmpS	Apply selected comparison operator to an array relative to a scalar
cvConvertScale	Convert array type with optional rescaling of the value
cvConvertScaleAbs	Convert array type after absolute value with optional rescaling
сvСору	Copy elements of one array to another
cvCountNonZero	Count nonzero elements in an array
cvCrossProduct	Compute cross product of two three-dimensional vectors
cvCvtColor	Convert channels of an array from one color space to another
cvDet	Compute determinant of a square matrix
cvDiv	Elementwise division of one array by another
cvDotProduct	Compute dot product of two vectors
cvEigenVV	Compute eigenvalues and eigenvectors of a square matrix



cvFlip	Flip an array about a selected axis
cvGEMM	Generalized matrix multiplication
cvGetCol	Copy elements from column slice of an array
cvGetCols	Copy elements from multiple adjacent columns of an array
cvGetDiag	Copy elements from an array diagonal
cvGetDims	Return the number of dimensions of an array
cvGetDimSize	Return the sizes of all dimensions of an array
cvGetRow	Copy elements from row slice of an array
cvGetRows	Copy elements from multiple adjacent rows of an array
cvGetSize	Get size of a two-dimensional array and return as CvSize
cvGetSubRect	Copy elements from subregion of an array
cvInRange	Test if elements of an array are within values of two other arrays
cvInRangeS	Test if elements of an array are in range between two scalars
cvInvert	Invert a square matrix



cvReduce	Reduce a two-dimensional array to a vector by a given operation
cvRepeat	Tile the contents of one array into another
cvSet	Set all elements of an array to a given value
cvSetZero	Set all elements of an array to 0
cvSetIdentity	Set all elements of an array to 1 for the diagonal and 0 otherwise
cvSolve	Solve a system of linear equations
cvSplit	Split a multichannel array into multiple single-channel arrays
cvSub	Elementwise subtraction of one array from another
cvSubS	Elementwise subtraction of a scalar from an array
cvSubRS	Elementwise subtraction of an array from a scalar
cvSum	Sum all elements of an array
cvSVD	Compute singular value decomposition of a two-dimensional array
cvSVBkSb	Compute singular value back-substitution
cvTrace	Compute the trace of an array
cvTranspose	Transpose all elements of an array across the diagonal
cvXor	Elementwise bit-level XOR between two arrays
cvXorS	Elementwise bit-level XOR between an array and a scalar
cvZero	Set all elements of an array to 0



That was only a little part for Matrix operations... !!!

There are also special methids that can be apply directly on an image (Smooth filtering, Canny, Hough transform, etc..)

http://www.seas.upenn.edu/~bensapp/opencvdocs/ref/opencvref_cv.htm

Here comes EMGU...

"Emgu CV is a cross platform .Net wrapper to the Intel OpenCV image processing library and allows OpenCv functions to be called from .NET compatible languages such as C#, VB, IronPython,..."

This means that it's possible to use OpenCV methods and structure in the C# simple style...



Example: the **IplImage** is defined in EMGU as an **Image** and is described (and instantiated since it's a class) by its generic parameters: color and depth

An image with 3 channels BGR each one defined by 1 byte:

Image<Bgr, byte> image=new Image<Bgr, byte>(new System.Drawing.Size(640, 480));

(The image will be managed by the garbage collector)

The main color types are supported :

Gray Bgr Bgra Hsv (Hue Saturation Value) HIs (Hue Lightness Saturation) Lab (CIE L*a*b*)



One of the most important method in EMGU is the **CvInvoke**, which allows to call directly the OpenCv functions (some OpenCV functions are wrapped in EMGU methods, but not all of them)...

IntPtr image = CvInvoke.cvCreateImage(new System.Drawing.Size(200, 200), Emgu.CV.CvEnum.IPL_DEPTH.IPL_DEPTH_8U, 1);

CvInvoke.cvDilate(ImageIn, ImageOut, myDilateElem, 1);

BUT.... For a basic list of methods that you can apply directly on the **Image**<**ColorType**, **Depht**> go:

http://www.emgu.com/wiki/files/2.3.0/document/Index.html

EMGU.CV.NameSpace -> Image (TColor, Tdepht) class -> Methods



- Emgu.CV Namespace
 - AdaptiveSkinDetector Class
 - AdaptiveSkinDetector.MorphingMethod En
 - CameraCalibration Class Capture Class

 - Capture.CaptureModuleType Enumeration ColorInfoAttribute Class

Collapse All

Code: All Members: Show All

- Contour(T) Class
- ConvolutionKernelF Class
- CvArray(TDepth) Class
- CvInvoke Class
- CvInvoke.CvAllocFunc Delegate
- CvInvoke.CvDistanceFunction Delegate
- CvInvoke.CvErrorCallback Delegate
- CvInvoke.CvFreeFunc Delegate
- EigenObjectRecognizer Class ExtrinsicCameraParameters Class
- FeatureTree Class
- HaarCascade Class
- HOGDescriptor Class
- HomographyMatrix Class
- ICapture Interface
- IColor Interface
- IConvexPolygon Interface
- IConvexPolygonF Interface
- IDuplexCapture Interface
- IDuplexCaptureCallback Interface
 ID
- IImage Interface
- Image(TColor, TDepth) Class
 - Image(TColor, TDepth) Members Image(TColor, TDepth) Constructor
 - Image(TColor, TDepth) Fields
 - Image(TColor, TDepth) Methods
 - Dilate Method
 - EqualizeHist Method
 - Erode Method
 - Flip Method
 - GammaCorrect Method

gu CV Library Documentation mage(<i>TColor, TDepth</i>) Methods image(<u>TColor, TDepth) Class</u> <u>See Also</u> <u>Send Feedback</u>		
//www.en	ngu.com	
mage(TC	color, TDepth) type exposes the following memb	bers.
lethods		
	Name	Description
\$	And	Inplace And operation with <i>src2</i> (Inherited from <u>CvArray(TDepth)</u> .)
\$	_Dilate	Dilates <i>this</i> image inplace using a 3x3 rectangular structuring element. Dilation are applied several (iterations) times
۹	<u>EqualizeHist</u>	The algorithm inplace normalizes brightness and increases contrast of the image. For color images, a HSV representation of the image is first obtained and the V (value) channel is histogram normalized
•	_Erode	Erodes this image inplace using a 3x3 rectangular structuring element. Erosion are applied several (iterations) times
\$	<u>Flip</u>	Inplace flip the image
\$	GammaCorrect	Gamma corrects this image inplace. The image must have a depth type of Byte.
\$	<u>Max(Double)</u>	Inplace compute the elementwise maximum value with <i>value</i> (Inherited from <u>CvArray(TDepth)</u> .)
\	<u>Max(CvArray(TDepth))</u>	Inplace elementwise maximize the current Array with <i>other</i> (Inherited from <u>CvArray(TDepth</u>).)
\	<u>_Min(Double)</u>	Inplace compute the elementwise minimum value (Inherited from <u>CvArray(TDepth)</u> .)
•	<u>_Min(CvArray(TDepth))</u>	Inplace elementwise minimize the current Array with <i>other</i> (Inherited from <u>CvArray(TDepth)</u> .)
\$	_MorphologyEx	Perform inplace advanced morphological transformations using erosion and dilation as basic operations.
\$	_Mul(Double)	Inplace multiply elements of the Array by <i>scale</i> (Inherited from <u>CvArray(TDepth)</u> .)
=	_Mul(CvArrav(TDepth))	Inplace elementwise multiply the current Array with src2

(Inherited from CvArray(TDepth).)



Let's see an example!!

