

General idea:

The main structure is generic image which is implemented to be similar to EmguCV's)

Differences and enhancements:

Generic image can be used in non-generic way also (useful for many filters and tools, see: some implementations in Accord.Imaging.Extensions project)

All operations (color conversions) are executed in parallel

Easier access to unmanaged data

All other projects are satellites around Accord.Imaging.Extensions.

Solution structure:

Demo – two demos

CamshiftDemo:

demo project for Camshift algorithm

Test:

demo project to demonstrate some Image extension functions (almost all functions are called as extensions) and performance

(you can see here some functionality on generic image)

Accord.Imaging.Extensions – **main project**

ColorSpaces

Definition of 3 color spaces (so far): *Gray.cs*, *Bgr.cs*, *Hsv.cs*

IColor.cs: interface for color, also a generic color spaces are defined there (Color3, Color4)

ColorAttribute.cs: defines color attribute, and the ColorInfo structure which is use in various functions (Generic image has property ColorInfo)

Extensions

AForgeCompatibility

AForgeFilterProcessing.cs: extensions for AForge filters processing

AForgeImageConversion.cs: GenericImage <-> UnmanagedImage (see Test)

ArrayCompatibility

GenericImage <-> Array extensions (see Test)

Basic

Extensions for image copying (with mask), flipping, range filtering, setting value

BitmapCompatibility

GenericImage <-> Bitmap extensions (see Test)

ComplexImage

Complex image arithmetic's extensions (Add, Mul, Div) + FFT

Convolution – only one extensions – Convolve (a type (spatial, FFT) and mode (sequential or parallel) are automatically chosen

FrequencyConvolution

FFT – FastFourierTransform (differs from AForge's) /not finished – (double implementation)/ – gives the same output as MATLAB's – (move to Accord.Math ?)

Complex.cs – (float and double Complex) - /not finished - members/ - (move to Accord.Math ?)

ParallelFFTConvolution.cs – parallel FFT convolution

SpatialConvolution

Drawing – drawing extensions

Math – math extensions (arithmetic's are located in core extensions)

Helper – helper methods and structures

HelperMethods.cs – methods for image copying and color -> array conversion

MethodCache.cs – methods for function caching – used for color type caching and generic image caching (move to Accord.Core ?)

PinnedArray.cs - an underlying object (buffer) for generic image (use in GenericImage)

SearchMethods.cs – Breadth first search – used for finding a color conversion path (see ColorConverter)
(move to Accord. Search algorithms?)

Image – Generic image

Core – because a generic image cannot be created from types (needed for some filters, a way to create generic image through types must exist -> *Image.cs*)

GenericImage.cs – generic image implementation (core)

IImage.cs – interface for image (if a user wants to use non-generic way)

Image.cs – generic image base – used to create a generic image from types when a standard way is not applicable)

CoreExtensions – extensions that are also used in GenericImage implementation (e.g. ChannelSplitter is also used in `im[0 /*ch idx*/]`)

ImageColorTypeConversions – there is just one extension: `Convert<TColor, TDepth>` (see `Image.ColorConversions`)

ParallelFilterProcessing

ParallelProcessor.cs – a core for parallel processing (splits 2D structure to patches where each patch is processed in parallel)
(move to Accord.Core ?)

Accord.Imaging.Filters.Extensions – filters for generic image /*not finished*/

Image filters (there should be extensions for each AForge and Accord filter)

(see Test (Sobel), Camshift (resize and smooth))

(move to Accord.Imaging.Filters ?)

Accord.Math.Geometry.Extensions – geometry structures and extensions

Some structures and extensions for already existing ones

(move to Accord.Geometry ?)

Accord.Vision.Extensions – vision algorithms

DenseHistogram – represents an image histogram (like EmguCV's) (move to Accord.Imaging)

Moments – your implementation (now is parallel and uses generic image, UnmanagedImage is also supported) (move to Accord.Imaging)

Tracking – Camshift and Meanshift trackers

(move to Accord.Vision – should replace existing implementations)