الجمهورية العربية السورية

جامعة دمشق

كلية الهندسة المعلوماتية

تقرير إنجاز وظيفة نظم الوسائط المتعددة في الهندسة المعلوماتية

PixelLab

Digital Image Processing & 3D Color Space Visualization Laboratory

العام الدراسي

2026 - 2025

الملخص التنفيذي

يقدم مشروع PixelLab بيئة عمل برمجية متقدمة لتحليل الصور الرقمية ومعالجتها وتمقيلها داخل فضاءات لونية متعددة بأسلوب تفاعلي ثلاي الأبعاد. يعتمد النظام على لغة C# ضمن بيئة .NET Framework. ويستخدم مكتبة Emgu CV بإعتبارها غلافاً متقدما لمكتبة OpenCV لتنفيذ العمليات الرياضية والمصفوفية عالية الكفاءة.

يدعم النظام التحويل الفوري بين عدة فضاءات لونية تشمل:

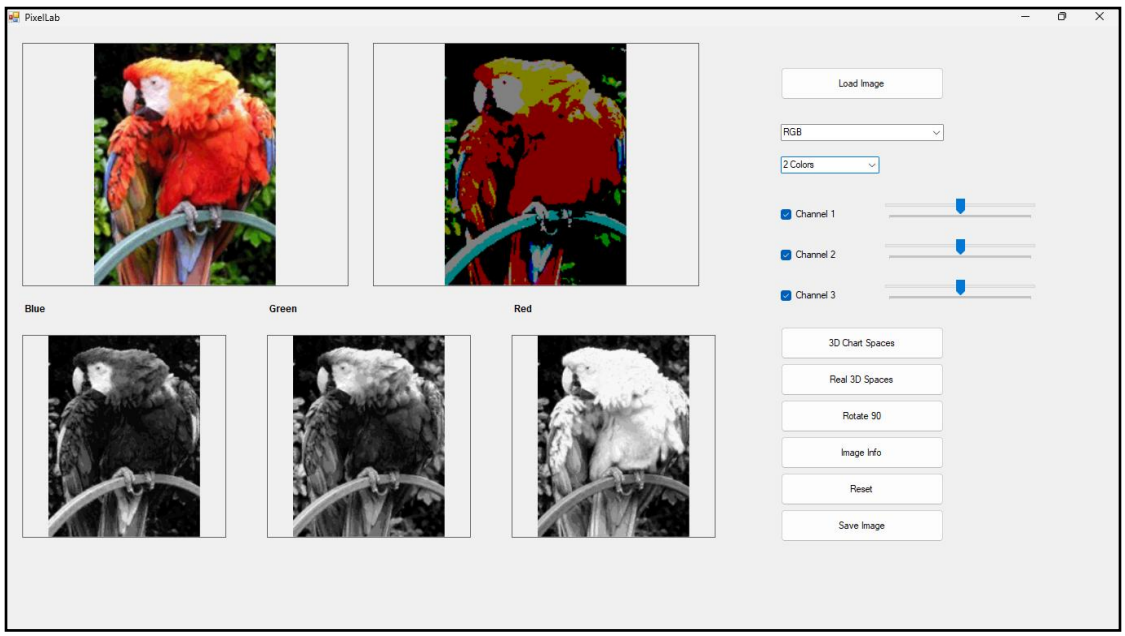
- RGB
- HSV
- LAB
- YCbCr
- CMYK
- YUV

كما يوفر أدوات تفاعلية للتحكم المباشر بقنوات الصور اللونية، وإجراء عمليات تحليل إحصائي للبكسلات بالإضافة إلى محرك رسوميات ثلاثي الأبعاد تم تطويره بالكامل من الصفر لتمثيل التوزيع اللوني داخل فضاءات هندسية مختلفة مثل:

- المكعب اللوني (RGB Cube)
- الأسطوانة اللونية (HSV Cylinder)
- المحاور الإدراكية (LAB Space)

ويتميز المشروع بإمكانية:

- تدوير المشاهد ثلاثية الأبعاد.
- التكبير والتصغير اللحظي.
- فحص أي بكسل وعرض قيمه في جميع الفضاءات اللونية.
- لمحاكاة الأنظمة الرقمية القديمة Color Quantization تنفيذ عمليات.



المقدمة والخلفية العلمية

تشكل معالجة الصور الرقمية أحد أهم فروع علوم الحاسوب والرؤية الحاسوبية، حيث تعتمد معظم الأنظمة

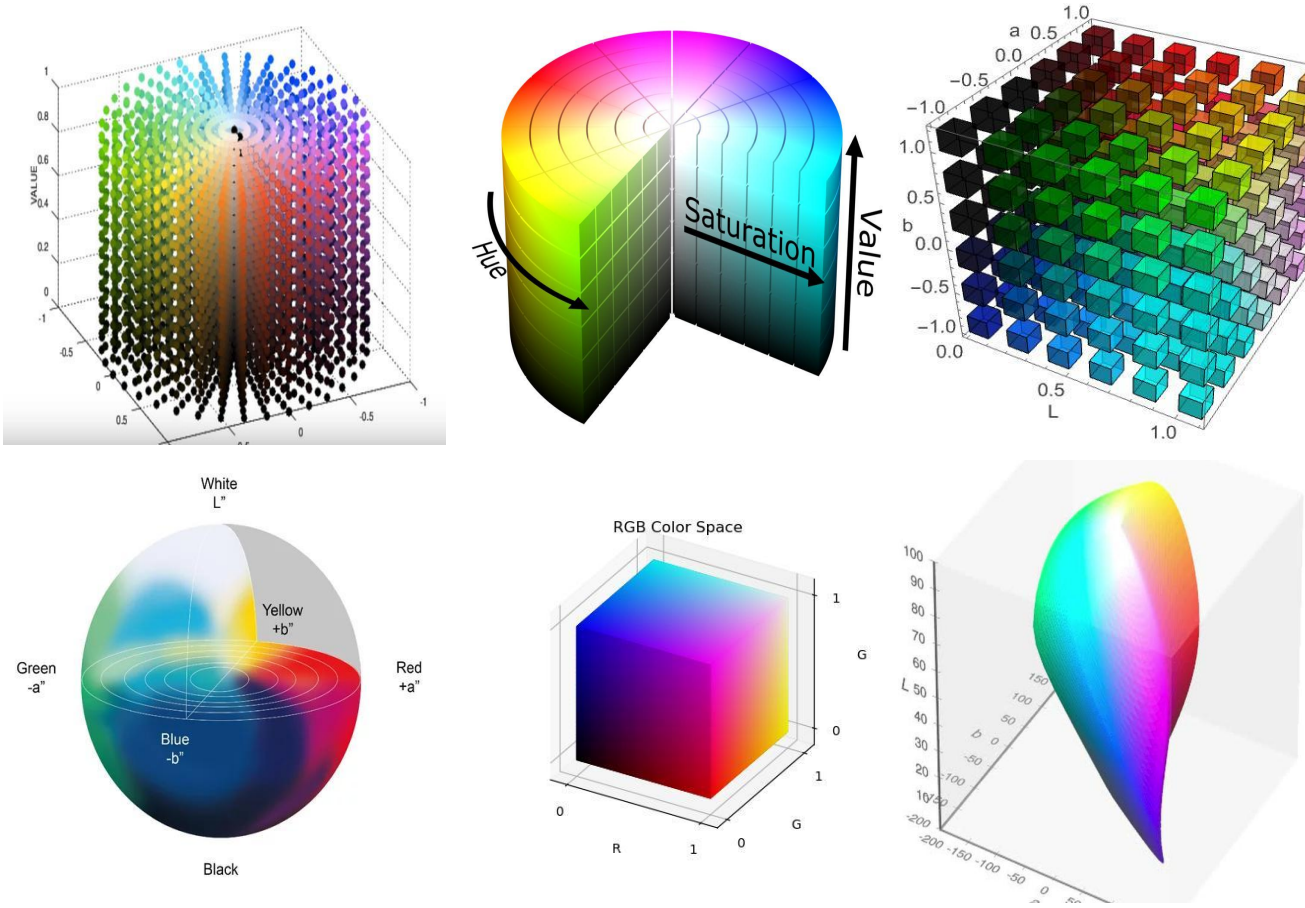
الحديثة مثل أنظمة الذكاء الاصطناعي، التعرف على الوجوه، المركبات ذاتية القيادة، والتطبيقات الطبية على تحليل الصور واستخراج المعلومات منها.

تعتمد الصور الرقمية على مفهوم البكسلات، حيث يحمل كل بكسل قيمة لونية يتم تمثيلها ضمن فضاء لوني معين. ولا يوجد فضاء لوني واحد مثالي لجميع التطبيقات، إذ تختلف الفضاءات اللونية بحسب طبيعة الاستخدام الرياضي أو البصري أو الصناعي.

فعلى سبيل المثال:

- يستخدم فضاء RGB في الشاشات وأنظمة العرض الرقمية.
- يسنخدم فضاء HSV لفصل اللون عن الإضاءة.
- يسنخدم فضاء YCbCr في ضغط الفيديو والصور.
- يستخدم فضاء LAB لمحاكاة الإدراك البشري للألوان.
- يستخدم فضاء CMYK في أنظمة الطباعة الاحترافية.

يهدف المشروع إلى بناء منصة تعليمية وبحثية تسمح بفهم الفضاءات عملياً من خلال التفاعل المباشر مع الصور وتمثيلها رياضياً وبصرياً.



أهداف المشروع

يهدف المشروع إلى تحقيق مجموعة من الأهداف العلمية والتقنية، أهمها:

الأهداف الوظيفية

- تحميل الصور الرقمية ومعالجتها.
- التحويل بين الفضاءات اللونية المختلفة.
- التحكم بقنوات الألوان بشكل مستقل.
- بناء تمثيل ثلاثي الأبعاد لتوزيع الألوان.
- تحليل القيم اللونية للبكسلات بشكل مباشر.

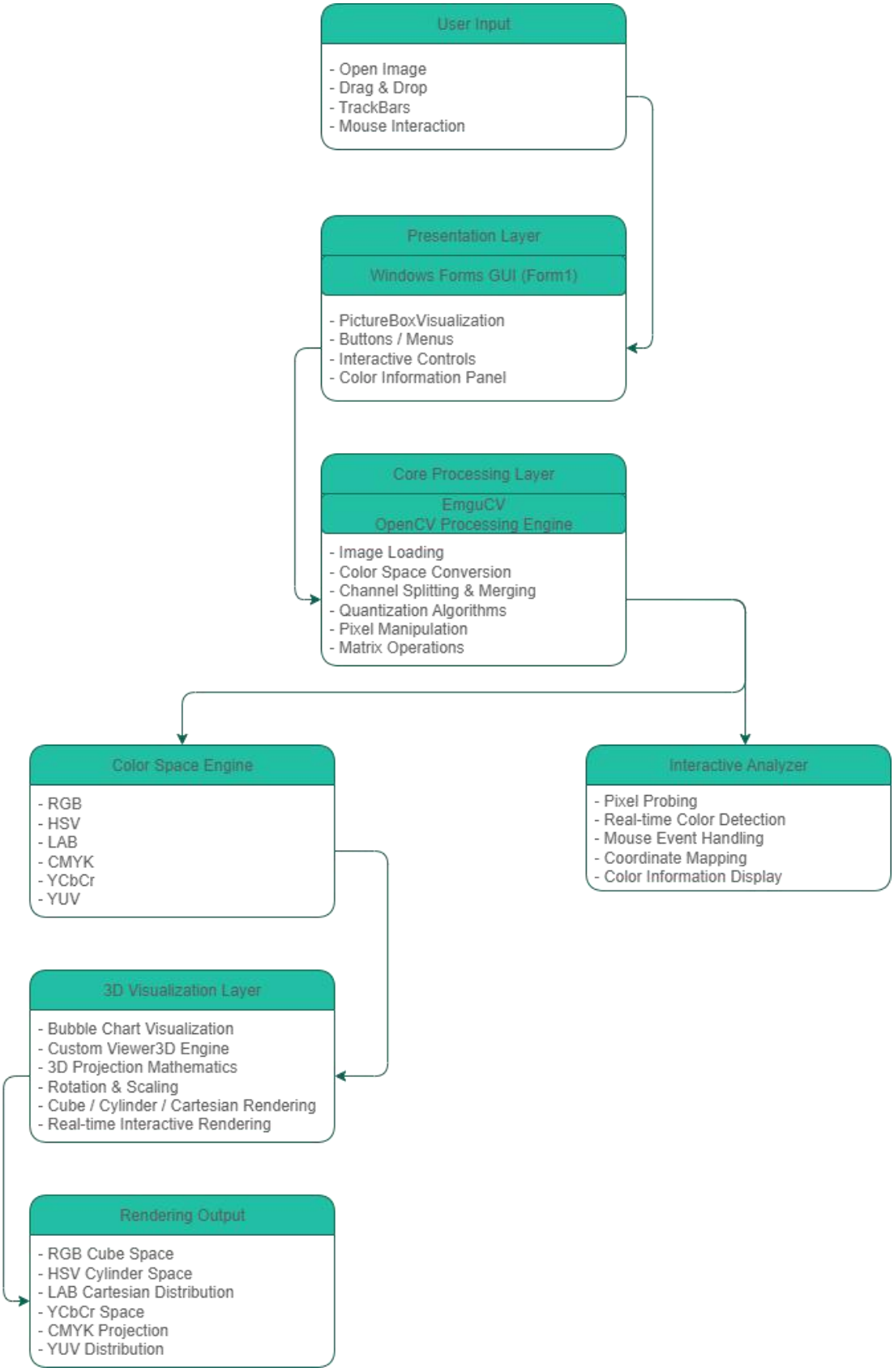
الأهداف البحثية

- دراسة التوزيع الإحصائي للألوان.
- مقارنة الفضاءات اللونية المختلفة.
- تحليل العلاقة بين الإدراك البصري والتمثيل الرياضي للألوان.
- دراسة تأثير تقليل العمق اللوني على الصورة.

التقنيات والأدوات المستخدمة

التقنية	الاستخدام
C#	تطوير التطبيق
.NET Framework	بناء واجهات النظام
Emgu CV	معالجة الصور
OpenCV	العمليات الرياضية
Windows Forms	الواجهة الرسومية
GDI+	الرسم ثلاثي الأبعاد
Mat & Image Classes	إدارة المصفوفات البيانية

Architecture Diagram



شرح المخطط المعماري

يوضح المخطط السابق البنية الطبقية الكاملة للنظام البرمجي PixelLab حيث يبدأ تدفق البيانات من طبقة إدخال المستخدم، المسؤولة عن استقبال الصور والأوامر التفاعلية المختلفة، ثم تنتقل البيانات إلى طبقة واجهة المستخدم الرسومية المبنية باستخدام Windows Forms.

والتي توفر أدوات عالية، EmguCV بعد ذلك، تتم معالجة الصور داخل طبقة المعالجة الأساسية باستخدام مكتبة الكفاءة لتنفيذ العمليات الرياضية والمصفوفية الخاصة بمعالجة الصور الرقمية وتحويلاتها اللونية.

تنقسم المعالجة لاحقاً إلى مسارين رئيسيين:

- محرك الفضاءات اللونية، المسؤول عن التحويل بين الفضاءات اللونية المختلفة مثل

RGB - HSV - LAB YCbCr - CMYK - YUV

- المحلل التفاعلي، المسؤول عن تحليل قيم البكسلات واستقبال تفاعلات المستخدم مثل النقر وتحريك الفأرة وتحديد الألوان.

وفي المرحلة الأخيرة، يتم تمرير البيانات إلى طبقة التصوير ثلاثي الأبعاد، والتي تعتمد على محرك رسوميات مخصص تم تطويره يدوياً لتنفيذ عمليات الإسقاط الهندسي والدوران والتكبير بشكل لحظي.

ينتهي النظام بإنتاج تمثيل بصري تفاعلي للفضاءات اللونية المختلفة، مما يسمح بدراسة التوزيع الإحصائي للألوان وتحليل العلاقات البصرية بينها بطريقة أكاديمية وعملية متقدمة.

البنية المعمارية للنظام

تم تصميم النظام وفق نمط Event-Driven Architecture، حيث تعتمد جميع العمليات على التفاعل مع المستخدم مع الواجهة الرسومية.

ينقسم النظام إلى ثلاث طبقات رئيسية:

طبقة إدارة الموارد (Resource Management Layer)

تتولى:

- تحميل الصور
- إدارة الذاكرة
- دعم السحب والإفلات (Drag & Drop)
- منع تسرب الذاكرة (Memory Leaks)

ويتم تخزين الصور ضمن الكائن Image<Bgr, byte>

طبقة المعالجة الأساسية(Core Processing Layer)

تتولى:

- فصل القنوات اللونية
- دمج القنوات
- تحويل الفضاءات اللونية
- تطبيق المرشحات والمعالجات الرياضية

ويتم الاعتماد بشكل رئيسي على:

```
CvInvoke.CvtColor()
CvInvoke.Split()
CvInvoke.Merge()
```

طبقة التصوير ثلاثي الأبعاد(3D Visualization Layer)

تتكون من جزأين:

أولاً: Bubble Charts

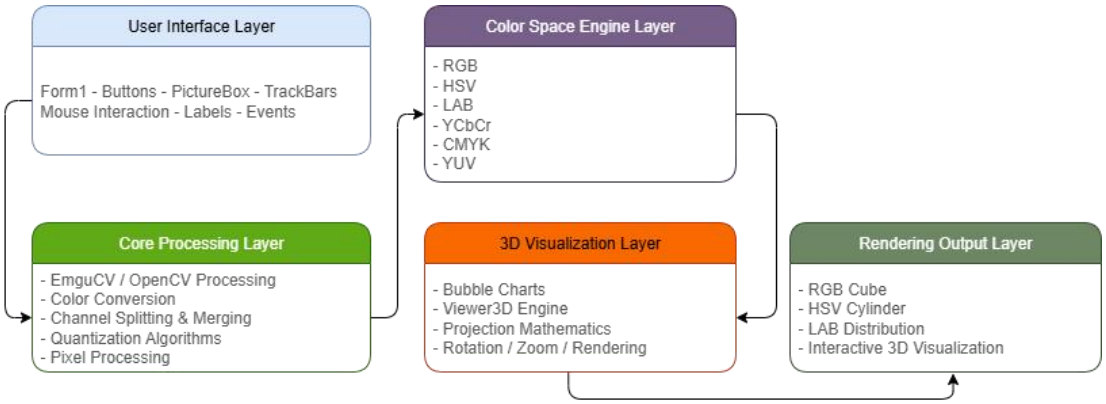
باستخدام:

System.Windows.Forms.DataVisualization.Charting

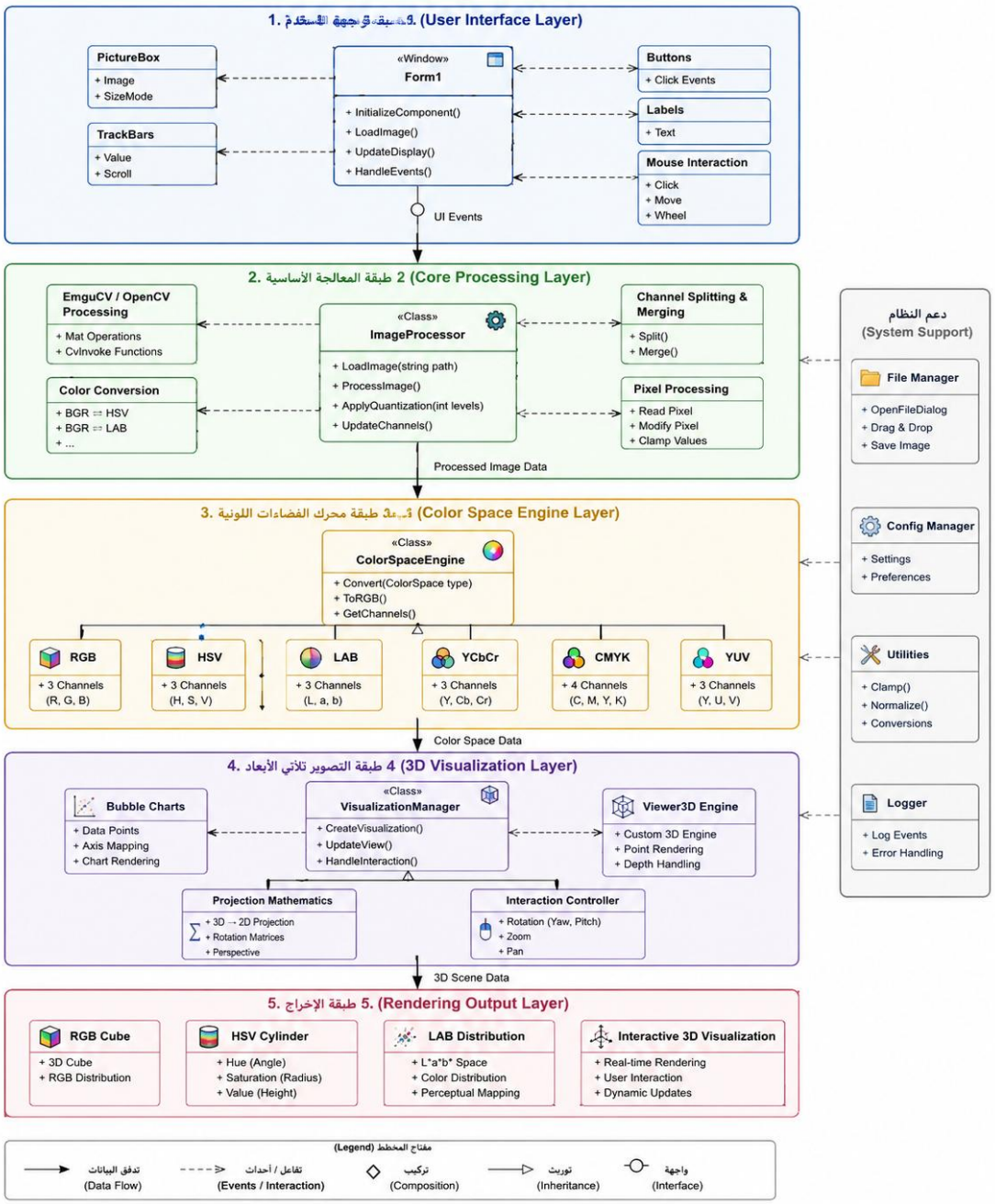
ثانياً: محرك Viewer3D

وهو محرك رسوميات مخصص تم تطويره يدوياً لإجراء الإسقاط الفراغي للنقاط

الطبقات البرمجية للنظام(Software Layered Architecture1)



Software Layered Architecture 2



يوضح الشكل السابق البنية الطبقية المعتمدة داخل النظام البرمجي، حيث تم تقسيم المكونات إلى طبقات مستقلة لتحقيق مبدأ الفصل البرمجي.

تبدأ العملية من طبقة واجهة المستخدم التي تستقبل تفاعلات المستخدم، ثم تنتقل البيانات إلى طبقة المعالجة الأساسية المعتمدة على Emgu CV لتنفيذ العمليات الرياضية والمصفوفية. بعد ذلك تتم معالجة الفضاءات اللونية داخل طبقة مستقلة، قبل إرسال النتائج إلى طبقة التصوير ثلاثي الأبعاد المسؤولة عن بناء المشاهد التفاعلية والإسقاطات الهندسية.

هذا التصميم الطبقي يجعل النظام أكثر قابلية للتوسع والصيانة وإضافة ميزات مستقبلية دون التأثير على باقي مكونات المشروع.

التحليل الرياضي والخوارزميات

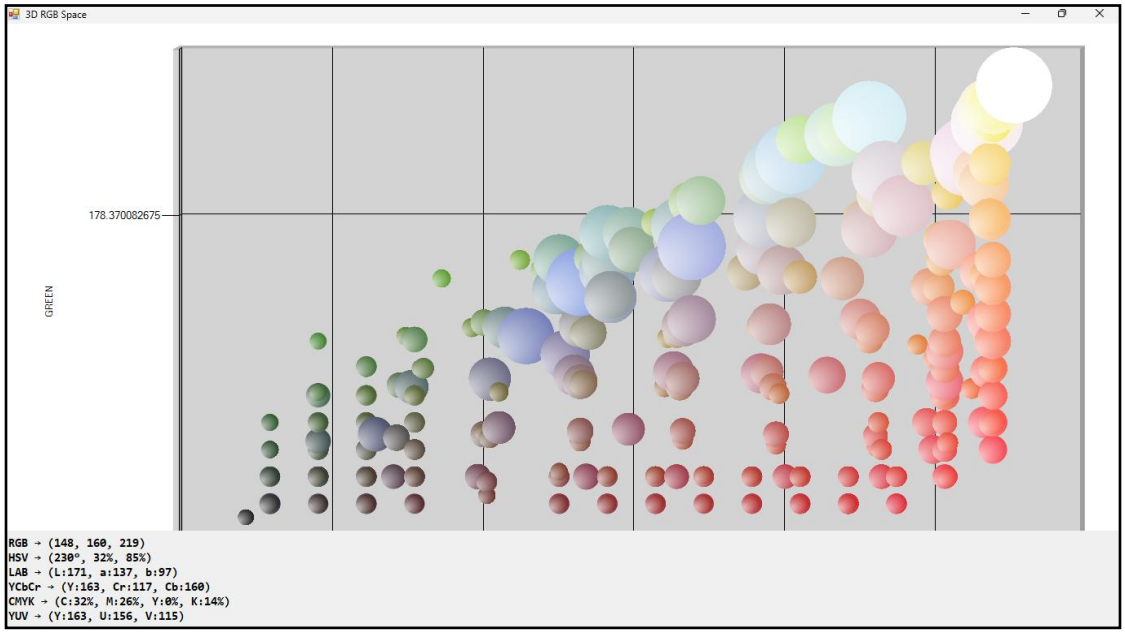
فضاء RGB

يمثل باستخدام ثلاثة محاور:

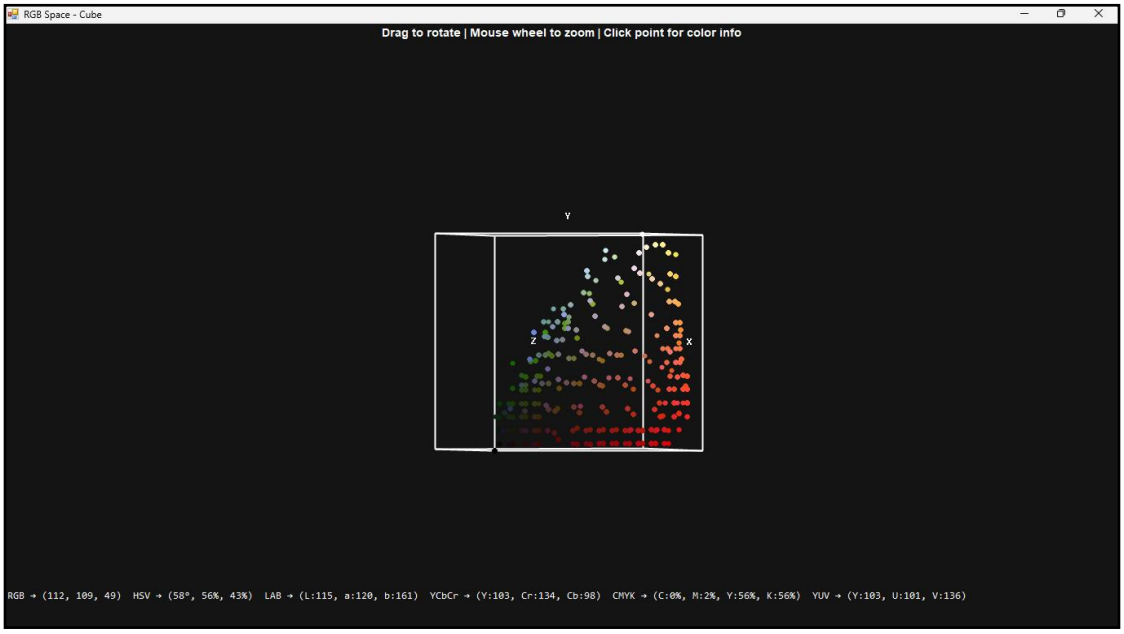
- Red
- Green
- Blue

ويتم تمثيله هندسياً على شكل مكعب ثلاثي الأبعاد.

3D Chart Space (RGB)



Real 3D Space(RGB)



فضاء HSV

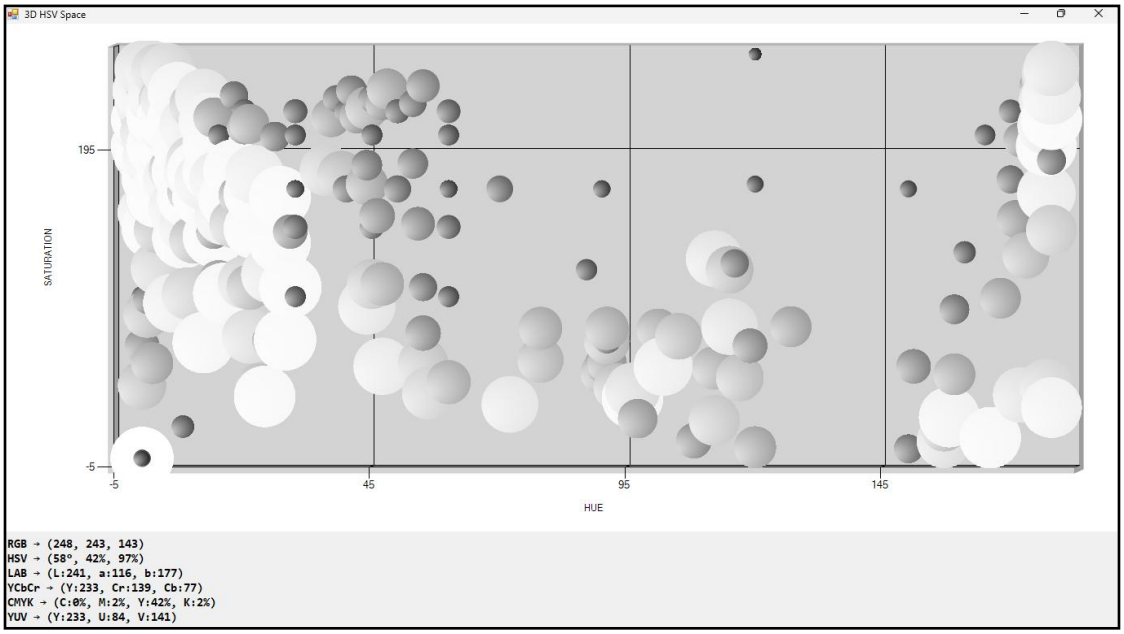
يعد هذا الفضاء من أكثر الفضاءات استخداماً في الرؤية الحاسوبية لأنه يفصل اللون عن الإضاءة .

حيث:

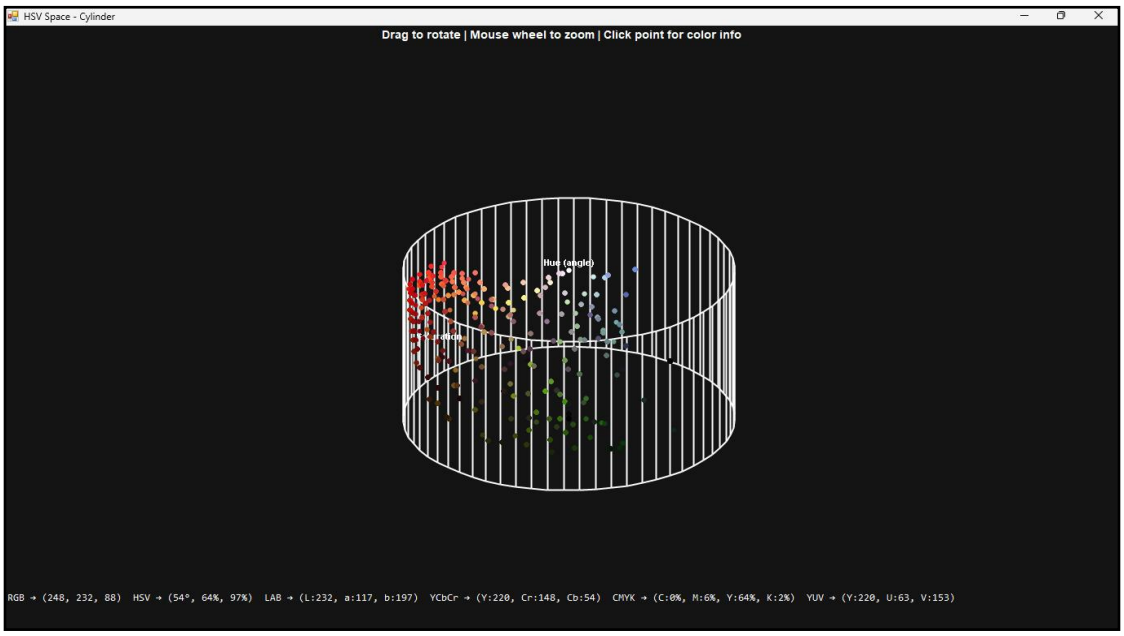
- Hue → زاوية لونية
- Saturation → شدة اللون
- Value → الإضاءة

العلاقات الرياضية المستخدمة: $X = S.\cos(H)$ $Y = S.\sin(H)$ $Z = V$

3D Chart Space (HSV)



Real 3D Space(HSV)



فضاء LAB

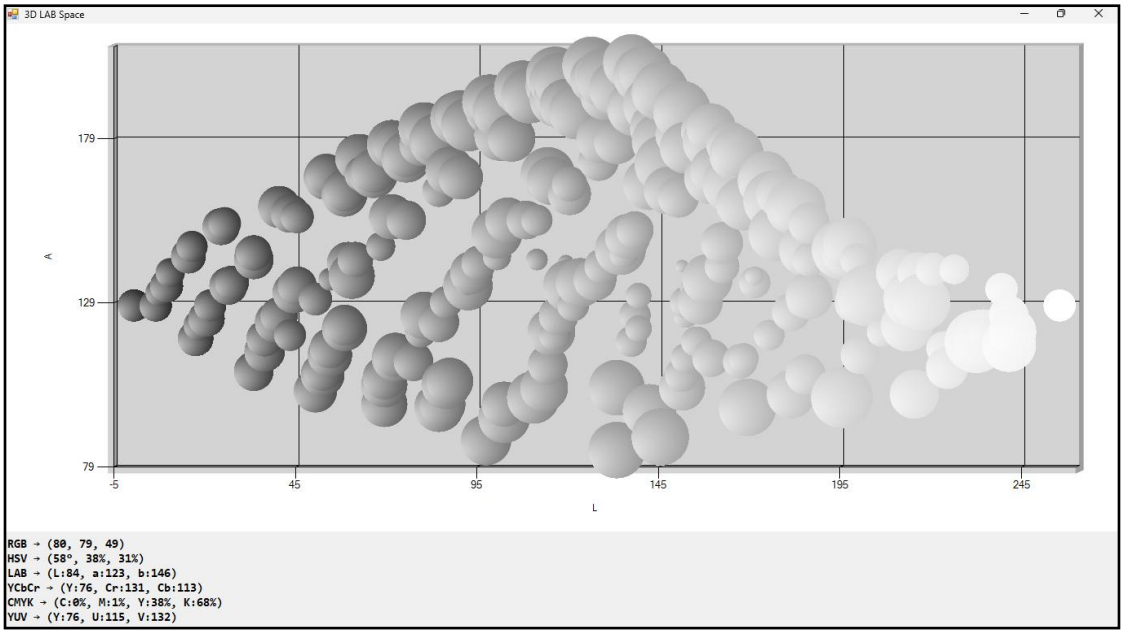
يحاكي هذا الفضاء الإدراك البشري للألوان.

ويتكون من:

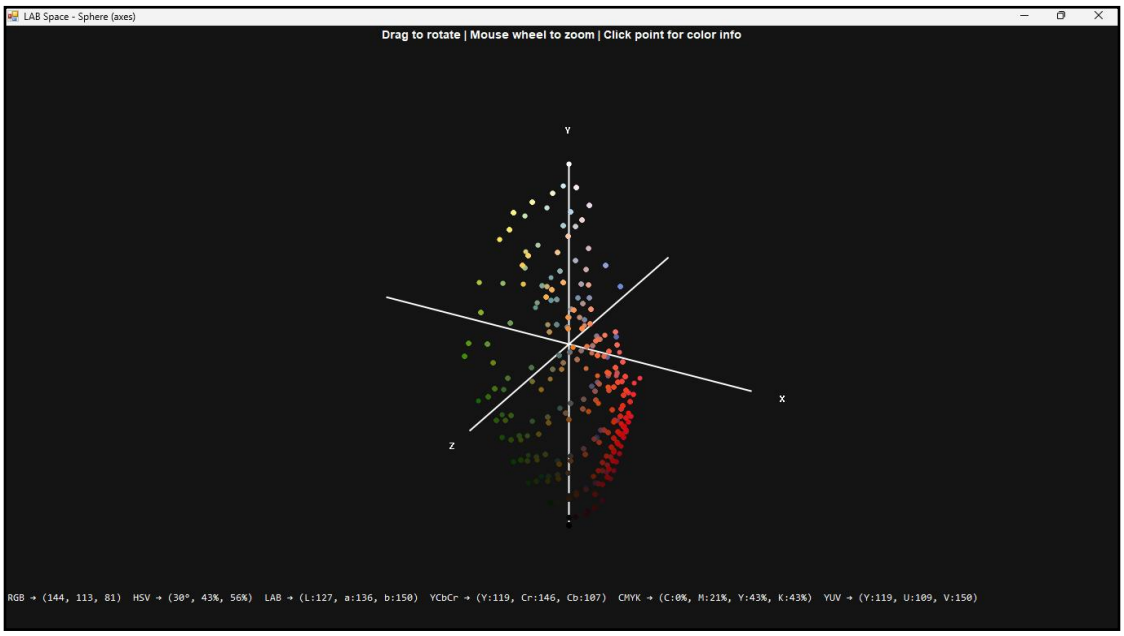
- الإضاءة → L
- المحور الأخضر/الأحمر → a
- المحور الأزرق/الأصفر → b

ويمتاز بدقة تمثيل الألوان مقارنة ب RGB.

3D Chart Space (LAB)



Real 3D Space(LAB)



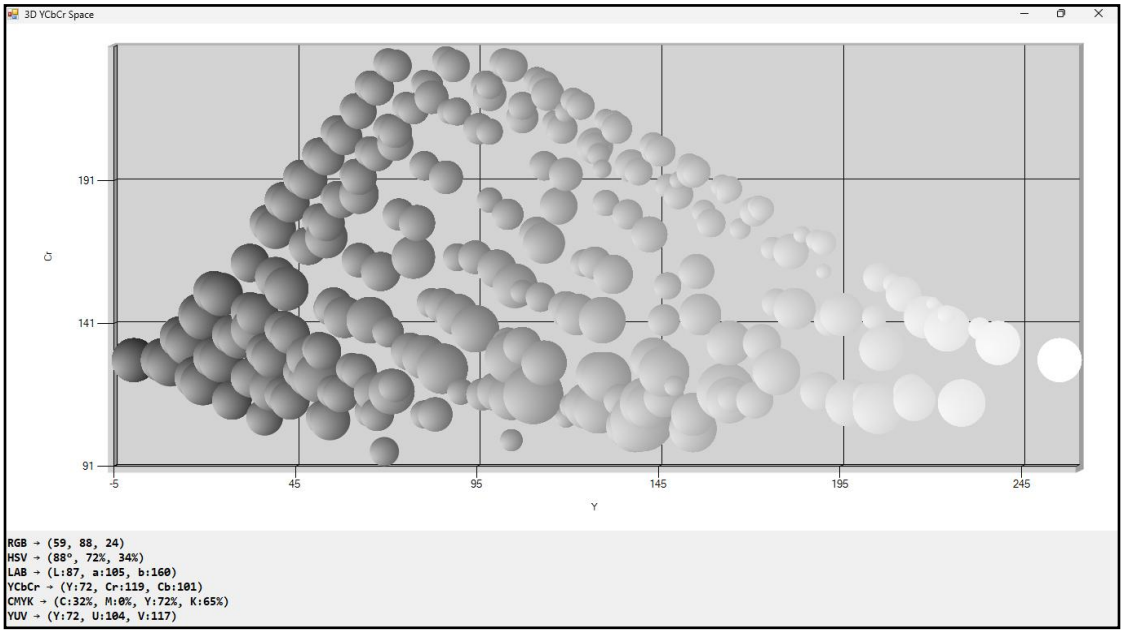
فضاء YCbCr

يفصل هذا الفضاء : الإضاءة (Luminance) عن التلوين (Chrominance)

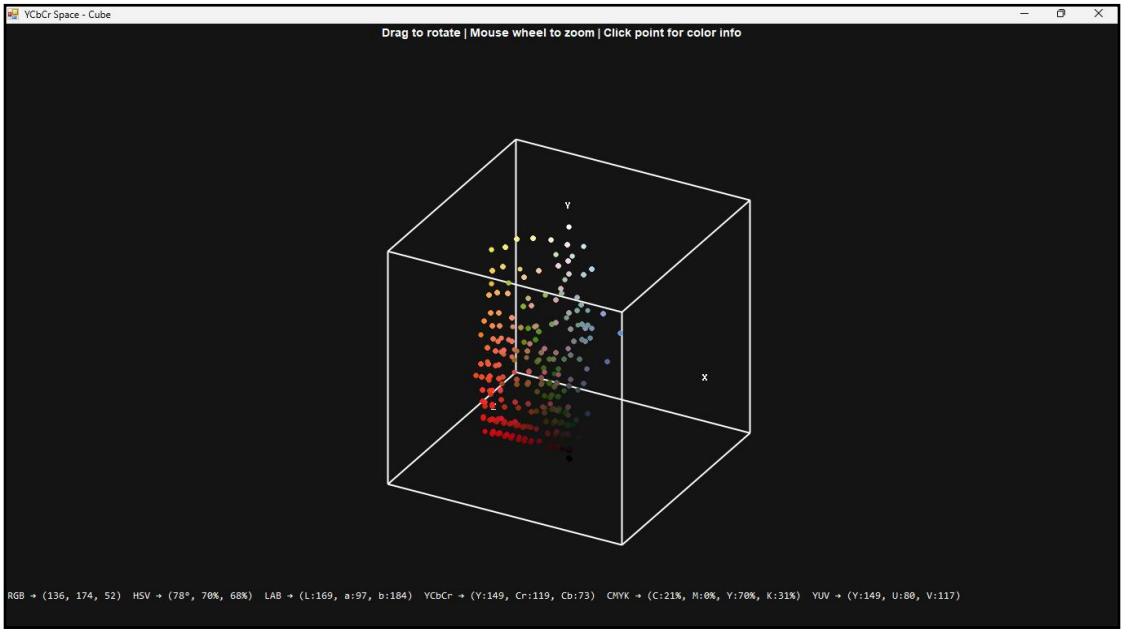
ويستخدم بشكل واسع في :

- JPEG
- MPEG
- ضغط الفيديو

3D Chart Space (YCbCr)



Real 3D Space(YCbCr)



فضاء CMYK

نظراً لأن OpenCV لا توفر دعماً مباشراً كاملاً لفضاء CMYK، تم تطوير الخوارزمية يدوياً.

أولاً يتم تطبيع القيم:

$$R'=255/R \quad ,G'=255/G \quad ,B'=255/B$$

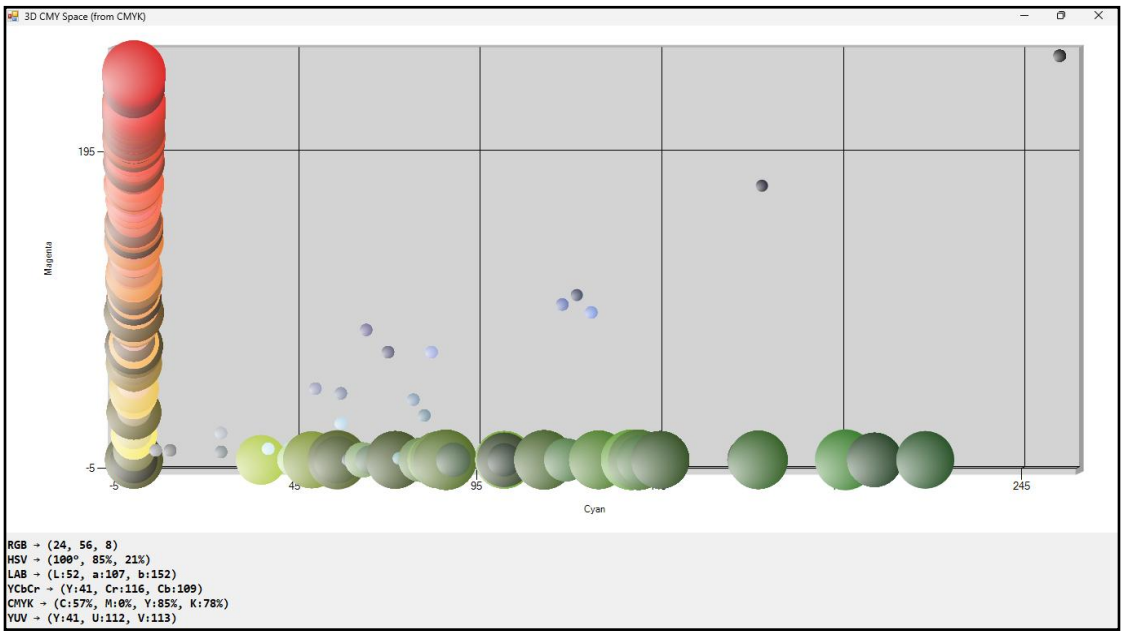
ثم حساب قناة الأسود:

$$K = 1-\max(R',G',B')$$

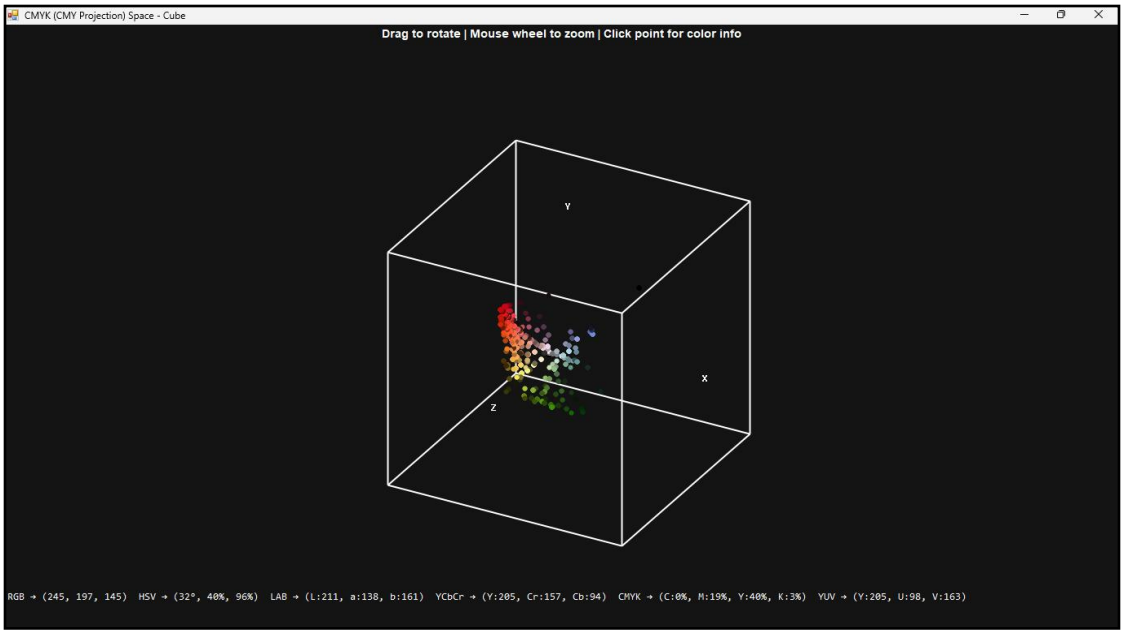
ثم استخراج القنوات:

$$C= (1-K1-R')/(1-K) \quad M= (1-K1-G')/(1-K) \quad Y= (1-K1-B')/(1-K)$$

3D Chart Space (CMYK)



Real 3D Space(CMYK)



فضاء YUV

نظراً لأن بعض المكتبات لا توفر تمثيلاً مباشراً موحّداً لفضاء YUV يتم تحويل القيم يدوياً من فضاء RGB باستخدام علاقات رياضية تعتمد على فصل الإضاءة (Luminance) عن اللون (Chrominance).

أولاً يتم استخدام قيم RGB كما هي أو بعد التطبيع إلى المجال من 0 إلى 1.

حساب مركبة الإضاءة Y (Luminance)

$$Y=0.299R+0.587G+0.114B$$

حساب مركبة اللون الأزرق U (Chrominance Blue)

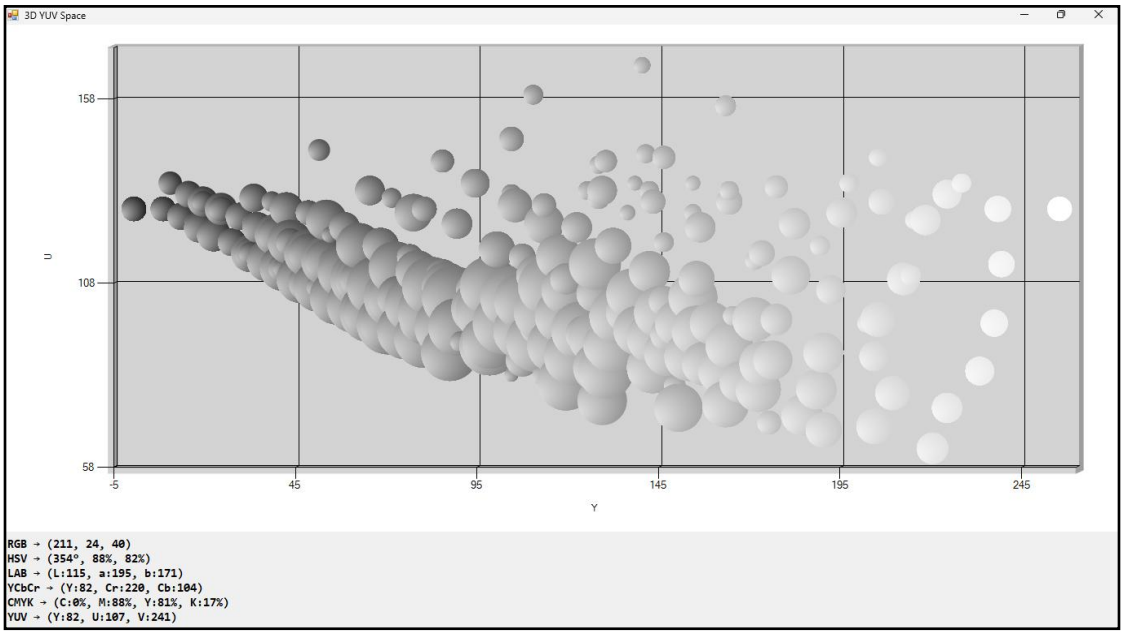
$$U=0.492(B-Y)$$

حساب مركبة اللون الأحمر V (Chrominance Red)

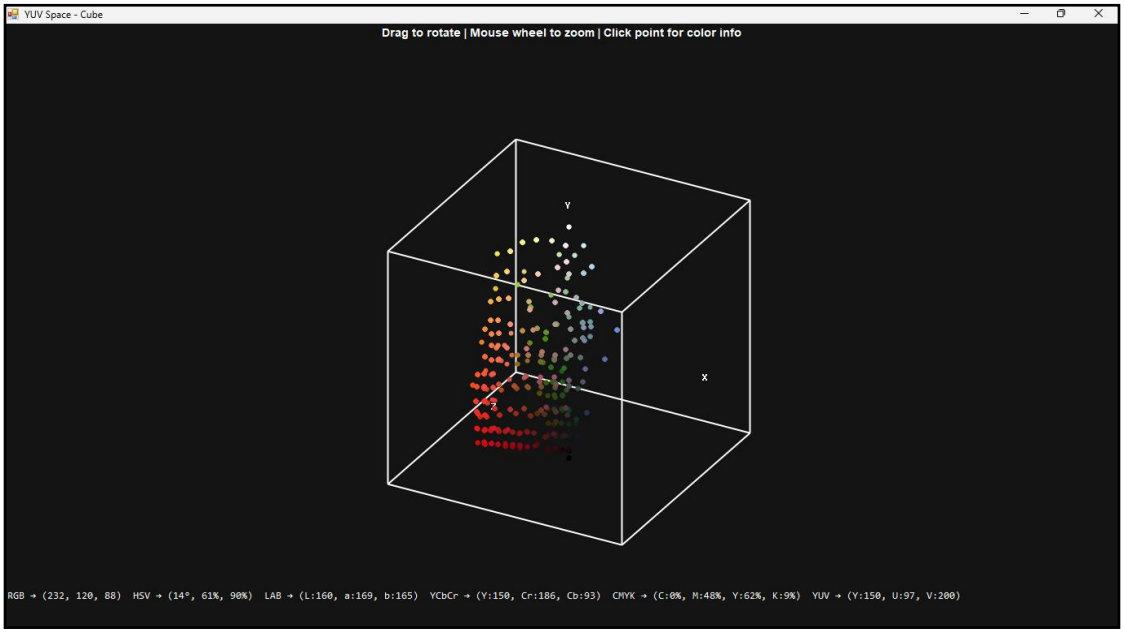
$$V=0.877(R-Y)$$

فضاء YUV يفصل الإضاءة عن المعلومات اللونية، لذلك يُستخدم بكثرة في ضغط الفيديو ومعالجة الصور .

3D Chart Space (YUV)



Real 3D Space(YUV)



خوارزمية تكميم الألوان (Color Quantization)

تهدف هذه العملية إلى تقليل عدد الألوان المستخدمة داخل الصورة.

يتم حساب خطوة التقسيم:

$$Step = 256/levels$$

ثم يتم تقريب القيم:

$$Q = (P/Step)\times Step$$

تؤدي هذه العملية إلى ظهور تأثير Posterization.

أمثلة عن تحديد الألوان للصورة:

8 Colors

4 Colors

2 Colors

256 Colors

16 Colors

شرح محرك الرسوميات ثلاثي الأبعاد (Real 3D Engine)

تم تطوير محرك Viewer3D يدوياً دون استخدام مكتبات رسوميات جاهزة .

يدعم المحرك:

- الإسقاط الفراغي
- الدوران
- التكبير والتصغير
- اختيار النقاط اللونية

الإسقاط الرياضي للنقاط

تعتمد عملية الإسقاط على مصفوفات الدوران المثلثية.

المعادلات المستخدمة:

$$x' = x\cos(\theta) - z\sin(\theta)$$

$$z' = x\sin(\theta) + z\cos(\theta)$$

التفاعل مع المستخدم

يدعم النظام:

- تدوير الفضاء عبر السحب بالماوس.
- التكبير عبر عجلة الفأرةز
- النقر على أي نقطة لإظهار قيمها اللونيةز

مثال من الكود:

```
yaw += dx * 0.01;
pitch -= dy * 0.01;
```

المسبار اللوني التفاعلي

عند الضغط على أي بكسل:

- يتم حساب موقعه الحقيقي داخل الصورة.
- يتم استخراج قيمه في جميع الفضاءات اللونية.

المعادلة المستخدمة:

$$Ximage = (Xmouse * Width(image)) / Width(control)$$

تم عرض سابقاً هذه الخاصية من خلال الضغط على أي بكسل داخل الفضائين لكل لون وعرض قيم البكسل بجميع الألوان.

تحليل الأداء

تم استخدام Sampling Step أثناء رسم النقاط لتجنب الضغط الكبير على المعالج :

مثال:

```
for (int y = 0; y < img.Height; y += 4)
```

يساعد ذلك على:

- زيادة سرعة الرسم
- تقليل استهلاك الذاكرة
- تحسين التفاعل اللحظي

الأداء	Sampling Step
بطيء	1
متوسط	2
ممتاز	4

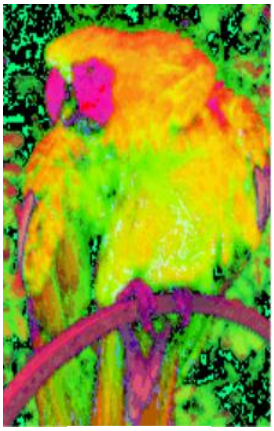
النتائج العملية (Experimental Results)

أظهرت التجارب النتائج التالية:

- توزيع RGB ظهر بشكل مكعب هندسي واضح
- فضاء HSV ظهر كأسطوانة لونية دقيقة
- فضاء LAB أظهر توزعاً أقرب للإدراك البشري
- تكميم الألوان أدى إلى تأثير Posterization واضح
- تعديل قناة Value في HSV سمح بتغيير الإضاءة دون تغيير اللون

التحويل بين الألوان:

RGB/CMYK



HSV

LAB

YCbCr

YUV

أيضا يوجد العديد من الخصائص مثل:

1. تدوير الصورة المعالجة
2. تكبير وتصغير الصورة المعالجة والأصلية
3. تفعيل وتعطيل قنوات الصورة مع إمكانية التعديل تأثير هذه القنوات على الصورة
4. عرض المعلومات الخاصة بالصورة المحملة
5. بعد القيام بعدة تعديلات عالصورة الأصلية حفظ التعديلات بالصورة المعالجة وحفظ الصورة بمسار يختاره المستخدم
6. يوجد زر عند الضغط عليه يقوم بإرجاع جميع التعديلات والعودة للصورة الأصلية

التحديات التقنية (Technical Challenges)

واجه المشروع عدة تحديات أهمها:

- معالجة ملايين البكسلات
- تحسين سرعة الرسم ثلاثي الأبعاد
- منع Memory Leaks
- منع Overflow و Underflow
- تحسين تجربة التفاعل اللحظي

الخاتمة

نجح مشروع PixelLab في بناء مختبر متكامل لمعالجة الصور الرقمية وتحليل الفضاءات اللونية وتمثيلها بصرياً داخل بيئة تفاعلية ثلاثية الأبعاد.

تميز المشروع بما يلي:

- دعم ستة فضاءات لونية
- تطوير محرك 3D مخصص
- تنفيذ خوارزميات رياضية دقيقة
- توفير تحليل بصري وإحصائي متقدم للألوان

ويمثل المشروع منصة تعليمية وبحثية قوية في مجال الرؤية الحاسوبية ومعالجة الصور الرقمية

المراجع (References)

- OpenCV Documentation
- Emgu CV Documentation
- Microsoft .NET Documentation
- Digital Image Processing — Gonzalez & Woods
- Computer Vision: Algorithms and Applications — Richard Szeliski