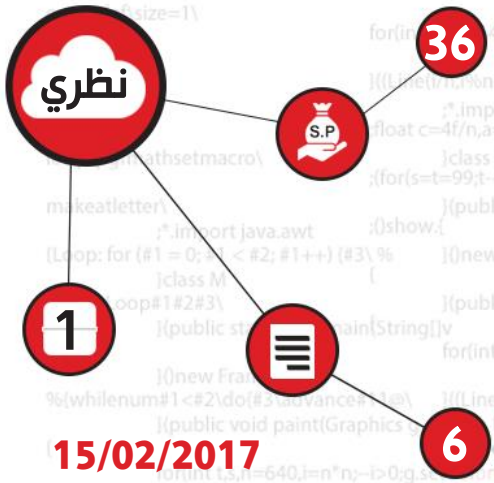


مقدمة

م.عمر سليمان

15/02/2017

نظم وسائط متعددة

أولاً - المفاهيم الأساسية التي سنتطرق لها في علوم الMultimedia :

1. النصوص.
2. الصور.
3. الصوت (الاصوات السمعية).
4. الفيديو.

بالإضافة الى ذلك يوجد الألوان.
و سنتعرف عليها وسندرس أنماط وأنظمة معينة منها.
وكيف يتم التحويل من نظام معين الى آخر.

مثال على أنظمة الألوان:

- RGB.
- CMY (وهو نظام يستخدم في الطباعة).

وسندرس أيضاً الضغط بالنسبة للصوت أو الفيديو
والخوارزميات المستخدمة في ذلك.

- **النتائج التي سنصل إليها من دراسة هذا المقرر:**

1. التعرف على هذه الوسائط.
2. كيف سيتم تمثيل هذه الوسائط.
3. المفاهيم الأساسية في الألوان.
4. المفاهيم الأساسية في الفيديو.
5. المفاهيم الأساسية في التحريك.

مثال:

التحريك بأن يتم تدوير الصورة Rotate او عكسها Mirror .

6. خوارزميات الضغط.

- لماذا نحتاج ال Media في حياتنا؟

البداية مع الـ Multimedia كانت من فترة ليست بقصيرة، فتصاعدت التقنية منذ ظهور الهاتف النقال و الانترنت و التلفاز الملون، والوسائط التي ساعدت على انتشار الـ Media وتطورها و رُقيها هي مواقع التواصل الاجتماعي.

فمثلاً :

- الكلام (الصوت) أسرع من الكتابة.
- السمع أسرع من القراءة.
- الرؤية اسرع من الوصف.

بالنسبة لكلمةMultimedia :

هي كلمة لاتينية تعني متعدد الوسائط ويقصد بها المشاركة وتوزيع المعلومات.

وهذه المعلومات مرقمنة ومضغوطة على الحاسب ويتم ارسالها بشكل رقمي.

كما نعلم ان الحاسب يخزن هذه الوسائط ويرسلها بسرعات مقبولة من مكان لآخر

أيضاً يوجد المصمم و المبرمج في هذا المجال وكل احد منهم قادر على عمل تحكم في هذه الوسائط عبر الحاسب

مثلاً:

- المبرمج قادر على التلاعب بالداتا (تغيير الوان صورة - تشفير نص وضغطه وتعديله - اقتصاص مقاطع من فيديو و الكثير من الخيارات)

-لماذا الرقمنة؟ او لماذا نرقمن الصور او الفيديو؟

* لاننا بحاجة الى جعلها بصيغة رقمية لتخزينها على الحاسب وعرضها ومعالجتها والتفاعل معها.

ملاحظة: التفاعل مع المحتويات الرقمية = Interactivity.

* ولاننا نريد نشرها على الويب يجب جعلها بصيغة رقمية.
* بالنسبة للفيديو لاننا نريد التحكم به بشكل افضل و الحصول على ميزة الـ Random Access وهي بان ننصل لاي مكان ضمن وقت الفيديو دون الانتظار و بشكل سريع.

مثال للتوضيح:

عند استماع لأغنية على شريط Caset ونريد الوصول لجزء معين فيها فأننا نحتاج للمرور على الشريط كله حتى نجد المقطع المطلوب اما عند رقمنة الأغنية فبنقرة واحدة نصل لأي مقطع نريده ضمنها.

تاريخ الMultimedia :

-بدايتها كانت بالجرائد عن طريق النصوص و الصور.

-في عام 1887 قام العالم توماس أديسون بعمل الصور المتحركة.

1895 - ظهر الإرسال الراديوي للإشارات اللاسلكية.

1930 - بعدها ظهر التلفاز وعنده بدأت القفزة في مجال الـ Multimedia.

1960 - بدأ ظهور الـ Hyper text و هي تمثل الصفحات التي يمكن التنقل ضمنها.

1967 - ظهر أول مخبر يعنى بالـMultimedia .

1968 - ظهر الـHyper Text Programmer .

1976 - ظهر نظام التجول في مدينة افتراضية.

1985 - بدأت الابحاث في مجال الـdigital video.

1989 - ظهر الـWorld Wide Web .

1990 - تأسس مخبر الـMultimedia lab .

1991 - ظهرت خوارزميات الضغط الرقمي.

1992 - ظهرت الخوارزميات التي تطورت عنها وهي JPG بتسلسلاتها.

1993 - ظهر اول متصفح للإنترنت.

1994 - ظهر اول محرك بحث.

1995 - ظهرت لغة الجافا والتي كانت تعنى بهواتف الـNOKIA .

1996 - ظهر الـ DVD Video وبعده ظهرت الـXML .

- و مازالـ هذا المجال يتطور حتى الآن.

تصنيفات الـMultimedia :

1. الإدراك بالرؤية والسمع.
2. الفورمات المستخدمة في تخزين الـ Multimedia

مثلاً :

- النص يشكل منASCII Code .
- الصورة تشكل من.OpenGL .

ما هي الـOpenGL؟

هي مكتبات يتم تضمينها في الـ C# او C++ وهذه المكتبات تعطي توابع تمكن المستخدم من رسم أشعة .Vector تستخدم في رسم دائرة بتعليمات الـ C# عن طريق براميترات هي (مركزها و نصف قطرها و خصائصها و الوانها مثلاً). وهذه الـ Vector ليست محصورة ضمن اشكال بدائية فيمكن تشكيل شخصيات مثل افلام الـ Animation و تضمينها واستدعائها والتحكم بها, عن طريق الـ OpenGL او الـDirectX .

-تدفق الصوت يمكن ان يعرض ضمنDCM .

والصور و الصوت يمكن أن يخزن ضمن صيغ معينة.

مصادر الـMultimedia :

الصورة يمكن انو تولد من لقطة لكاميرا او رسم ضمن الحاسب يدوي.

و كذلك الأمر بالنسبة للفيديو من ناحية الإدخال.

-لاتقاط الصورة من Scanner اول مرحلة تمر فيها الصورة اسمها الـ Sampling وبعد ذلك يتم الحصول على مشهد رقمي.

-بعد تحويلها لشكل رقمي يتم تحديد مكان تخزينها أي على القرص الصلب اوCD .

ثانياً - يجب التمييز بين الأمواج المستمرة و الأمواج الرقمية، وتمييز الأبعاد:

1. الصوت: الصوت يأخذ بعداً واحداً.
2. الصورة: الصورة تمثل ببعيدين، الأول للطول و الثاني للعرض.
3. الفيديو: الفيديو يمثل بثلاث أبعاد، إثنان للـ لقطات و واحد للزمن و الصوت.
4. الرسوم: الرسوم تمثل بأربع أبعاد, ثلاثة للفيديو وواحد للزمن.

مسألة:

يوجد لدينا صفحة فيها 60 سطر و كل سطر فيه 80 محرف وكل محرف يمثل بـ 8 بت, أحسب حجم هذه الصفحة بالكيلو بايت.

الحل:

حساب عدد الأحرف: 80 × 60 = 4800 محرف

المساحة التي تشغلها الأحرف بالبت: 4800 × 8 = 38400 بت

تحويل إلى بايت: 38400 ÷ 8 = 4800 بايت

تحويل إلى كيلو بايت: 4800 ÷ 1000 = 4.8 كيلو بايت

- **شرح عن الـ Vector والـ Raster :**

الـ **Vector**: معنى هذا الكلمة شعاع, ونحصل عليه نتيجة تعليمات معينة في الـ **OpenGL** كما ذكرنا.

الـ **Raster**: هي نتيجة التقاط صورة من كاميرا مثلاً, ويكون مجموعة من النقاط اللونية أو الـ **Pixel**.

- **الفرق بينهما:**

Vector	Raster
يتشكل من خلال أكواد وتعليمات برمجية.	يتشكل من خلال صورة من كاميرا مثلاً.
يتطلب مساحة تخزينية قليلة. **بسبب تكونه من الأكواد الكتابية.**	**يتطلب مساحة تخزينية عالية.** **بسبب التفاصيل التي تحتاجها كل نقطة من الرسم.**
يمكن التحكم بالأشكال بكل بساطة من خلال تعديل البرامترات للرسم.	يصعب التحكم بالأشكال داخل المشهد بسبب عدم وجود قاعدة لتشكل التفاصيل فيه.
مهما حاولنا تحسين الصورة او المشهد وتجميله لا يمكن جعله واقعي.	**بكل بساطة من خلال التقاط صورة يمكنها أن تمثل الواقع بشكل ممتاز.**
يسهل تشكيل الأشكال فمثلاً يمكن رسم دائرة بكل بساطة وإعطائها لون فتتلون بلون واحد.	يصعب تشكيل الأشكال فمثلاً رسم الدائرة يتطلب إعطاء كل نقطة منها لون لكن هنا يمكن تلوينها بعدة الوان.
لا توجد هذه المشكلة تقريباً. **وإن ظهرت يمكن حلها بكل سهولة من خلال المعادلات.**	**نعاني من مشكلة الدرج المتكسر.** **تظهر هذه المشكلة عند تقريب الصور.** **يصعب حلها.**
يمكن تعديل الحجوم داخل الصورة الواحدة للأشكال الموجودة وتحديدها بكل سهولة.	يصعب علينا تعديل او تحديد أي عنصر داخل الرسم الواحد.
الصور المعقدة تستهلك وقت كبير لإخراجها على الحاسب.	**يستهلك وقت قصير في إخراج الصور المعقدة**
يحتاج حواسيب بمواصفات عالية للتعامل معها بشكل أسهل وأسرع (عند التعقيد).	لا تتطلب حواسيب بمواصفات عالية. يتطلب وقت أقل في الإظهار على الحاسب
لا تدعمها جميع المتصفحات على الويب.	**يتم دعمها بشكل عام.**
لا يتشوه عند تعديل الحجم.	يتشوه إذا قمنا بأي تعديل على الحجم.

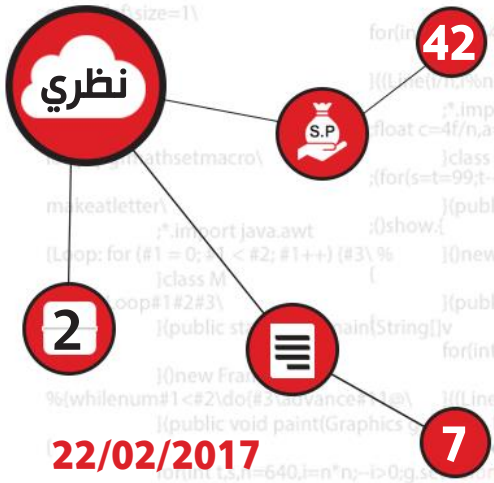
- **الفيديو:**

هو نفس الصورة بالخصائص ولكن مع وجود عامل الزمن وتعديلات تطرأ على الصورة خلال الزمن.

الـFrame: نقصد به عدد النقاط اللونية بالطول والعرض, الـFrame هو الصورة التي تظهر بالفيديو ويكون له نسبة مثلا 30 FramePerSec أي سيتم عرض 30 إطار خلال الثانية الواحدة داخل الفيديو.

- **التفاعلية بالـ Multimedia:**

1. عند بث فيديو على الويب: تحقيق إمكانية الإنتقال لأي نقطة يريدها المستخدم داخل الفيديو.
2. التسامح مع الأخطاء: عند إرسال الفيديو بنسبة 25 FPS ولسبب ما ضاع عدد من الـلقطات عند الإرسال تحدد الخصائص داخل الـ Multimedia طريقة التعامل مع أخطاء كهذه.



الألوان
(في الصورة)

م.عمر سليمان

22/02/2017

RBTCC#

نظم وسائط متعددة

أنواع الالوان وأسباب تطويرها:

- RGB: جيد للتعامل مع الصور على الحاسب وللعرض على جميع انواع الشاشات.
- CMY: جيد للطابعات والراسمات.
- CMYK: مثل ال CMY ولكن يسهل التعامل مع الألوان المعتمة بسبب إضافة اللون الأسود لهذا النظام.
- HSB: نظام خاص جيد للمستخدم والتعامل مع الألوان الحقيقية للعين البشرية.
- HSV: مثل ال HSB.

نظام الألوان RGB:

- يستخدم لتلوين رسومات ال Raster وللعرض على شاشات المدفع CRT.
- يوظف هذا النظام في تمثيل الجداء الديكارتي (**أي انه يمثل على ثلاث محاور إحداثية**).

ومن خلال التعامل مع هذا التمثيل ينتج اللون وقيمه تكون على شكل مكعب

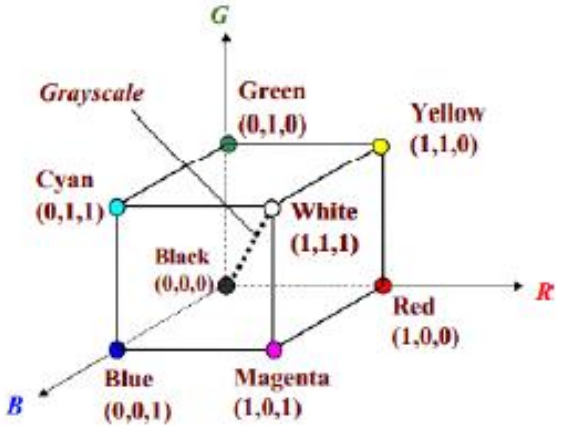
- القيم اللونية لهذا النظام محضورة بين (0, 1) أي أن المجتزء الأول R إما يكون 0 او 1.

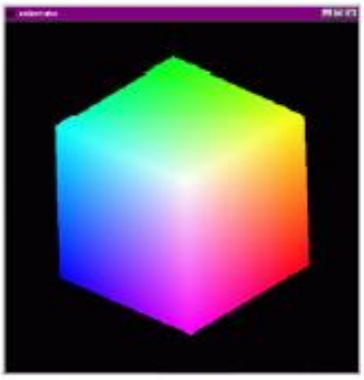
- يسمى هذ النظام أيضاً بنظام الألوان الجمعي(Additive primaries color system).
- الألوان في الفيديو والصورة تنتج عن جمع كميات من الألوان الثلاثة:

$$C_{RGB} = rR + gG + bB$$

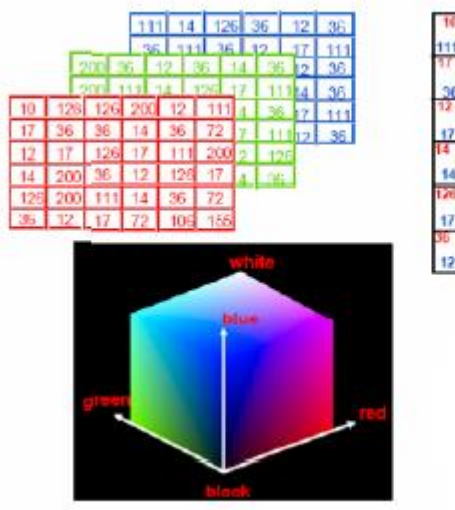
عندما تكون ال r g b تمثل كميات من الألوان الأحمر والأخضر والأزرق.

- الإحداثيات هي الألوان الرئيسية لهذا التمثيل ورؤوس المكعب على هذه الإحداثيات هي القيم العظمى والرؤوس المتبقية هي الألوان المكملة لكل لون أساسي.





RGB color cube



يتم تمثيل الألوان في الشاشات لكل نقطة على حدى ويختلف هذا التمثيل من شاشة لأخرى

- من الملاحظ أن تدرجات اللون الرمادي هو القطر بين الأبيض والأسود في المكعب أي بين النقطة (0, 0, 0) والنقطة (1, 1, 1).
- جميع نقاط التدرج الرمادي تتساوى بالكميات من ناحية جميع الألوان الباقية RGB.
- عند تحويل صورة من الألوان الطبيعية إلى تدرجات الرمادي فإننا سنقوم بإستخدام برنامج يقوم بجعل جميع قيم RGB متساوية لكل نقطة داخل الصورة.
- العين البشرية حساسة بشكل اكبر للون الأخضر وحساسة بشكل قليل للون الأزرق.

تمثل هذه الحساسية بقيم :

$$L = 0.30R + 0.59G + 0.11B$$

إذا كان C2 + C1 = لون أبيض

فهذا يعني أن اللونين C1 و C2 متكاملين

مثال على الوان متكاملة:

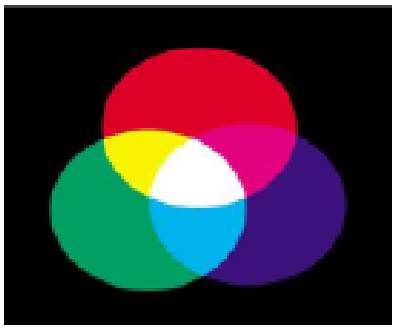
Red & Cyan

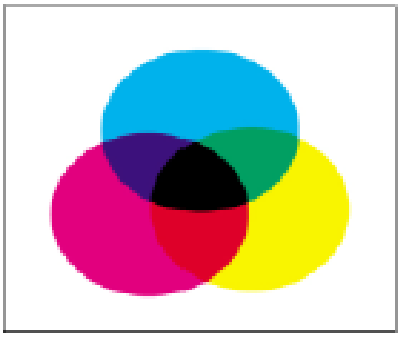
Green & Magenta

Blue & Yellow

نظام الألوان CMY:

- تستخدم في الطابعات ذات الأربع الألوان (اللون الرابع هو الأسود), وتحدد الألوان في كمية الحبر المسخدم.
- مطروح الألوان الأساسية (الألوان الغامقة): يتم تحديده من قبل ما هو مطروح من اللون الأبيض وليس من خلال ما يضاف للون الأسود.
- الأزرق السماوي(Cyan) هو ناتج عن طرح اللون الأحمر من الأبيض وجميع الألوان في هذا النظام تنتج عن الطرح من الأبيض او عن دمج ألوان نظام ال RGB.





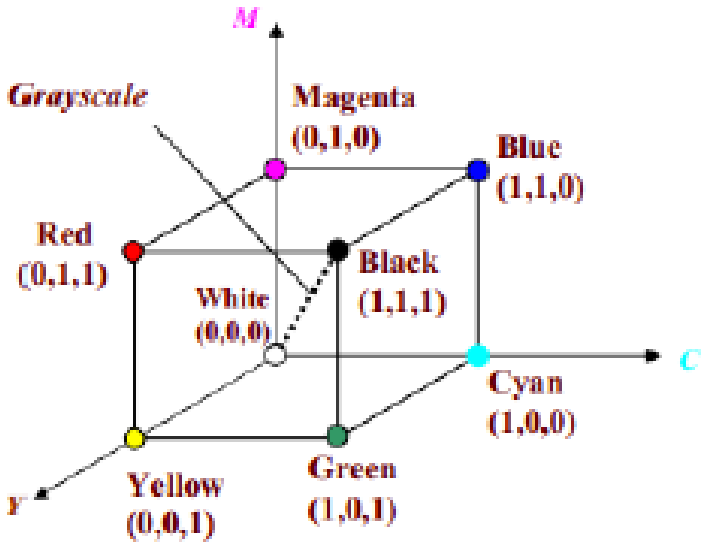
C = W - R = G+B

M =W - G = R+B

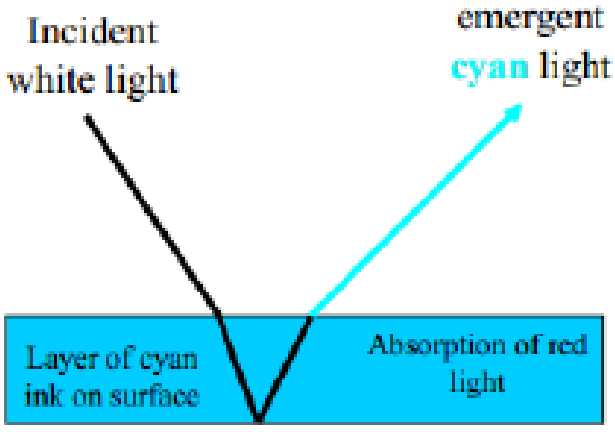
Y =W - B = R+G

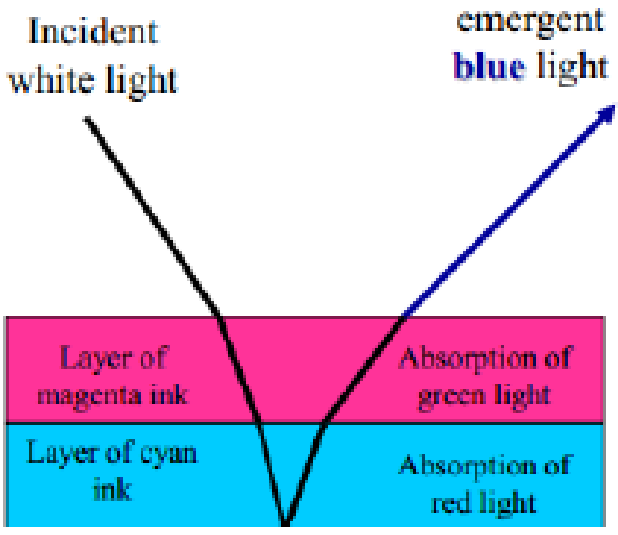
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



- يستخدم هذا النظام في الطابعات النافثة للحبر والراسمات التي تضع الصباغ على الورق.
  - عندما يطلى السطح بلون ازرق سماوي لا ينعكس اللون الأحمر من الإضاءة فيظهر للعين اللون الأزرق السماوي.
  - عندمى يطلى سطح الورقة باللونين الأرجواني والأزرق السماوي فإن اللون الأحمر و الأخضر لا ينعكسون عن سطح الورقة فترى العين اللون الأزرق.



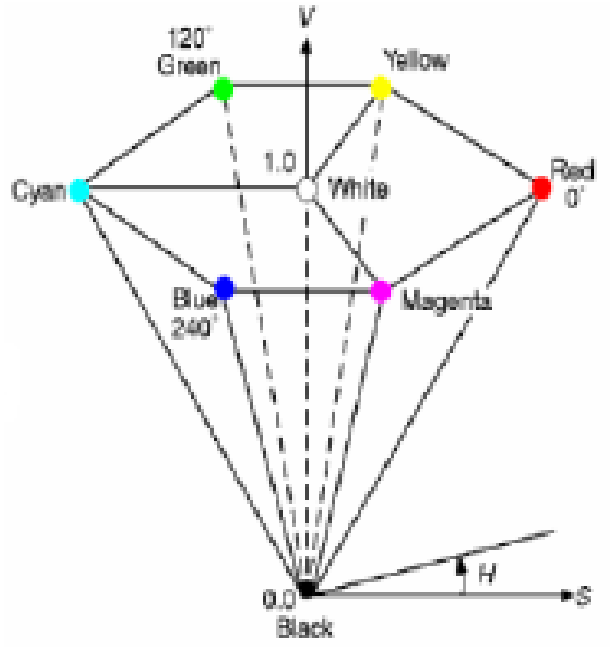


نظام الألوان CMYK:

- هو نفس التنسيق السابق لكن مع إضافة اللون الأسود كـ لون اساسي.
- بمزج اللون الأسود مع الألوان الأساسية لهذا التنسيق نحصل على ألوان اغمق, لأن اللون الأسود يمتص اللون الأبيض الموجه نحو الرسم او النقطة اللونية المضاف لها هذا اللون.
- خلط الألوان الساسية لهذا التنسيق ينتج لدينا الألوان الساسية لتنسيق الRGB وبإمكاننا الحصول على كافة التدرجات من الألوان الأساسية للتنسيقين معا من خلال الدمج.
- جمع حبر الألوان الثلاثة لهذا التنسيق لا ينتج لنا لون اسود جيد.

الفضاء اللوني HSB او HSV :

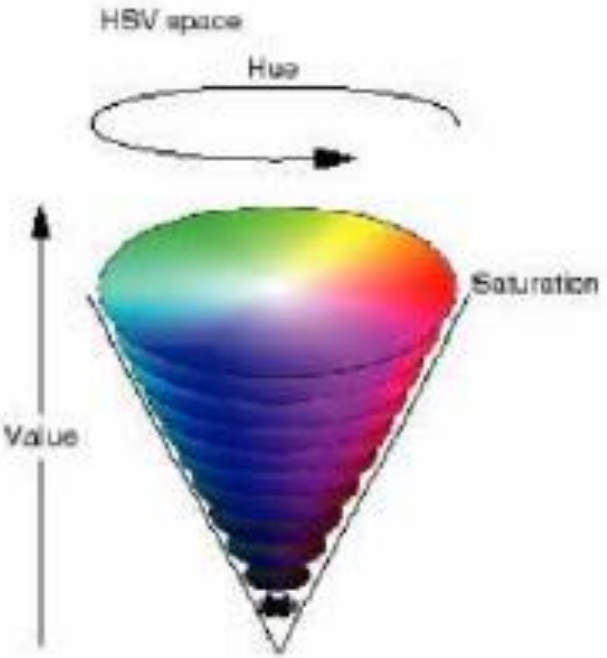
- هو ليس نظام لوني مستقل وإنما هو تنسيق خاص للألوان لمقاربتها أكثر للعين البشرية.
- HSV مصمم لمحاكاة الطريقة التي ترى بها العين البشرية الألوان وتتفاعل معها.
- هذا التنسيق هو تحويل نظام الRGB إلى الوان طبيعية أكثر ليتعامل معها الفنانين او الأدوات المخصصة للتصميم.
- HSV هي إختصار (Hue, Saturation, Value) معناها التدرج اللوني و الإشباع و القيمة.
- HSB هي إختصار (Hue, Saturation, Brightness) معناها التدرج اللوني و الإشباع و السطوع.
- يتمثل هذا التنسيق ضمن شكل مخروطي.



- السطح الأفقي الأعلى من المخروط مقسم لشكل سداسي الأضلاع وكل زاوية منه تمثل لون واندماجاته

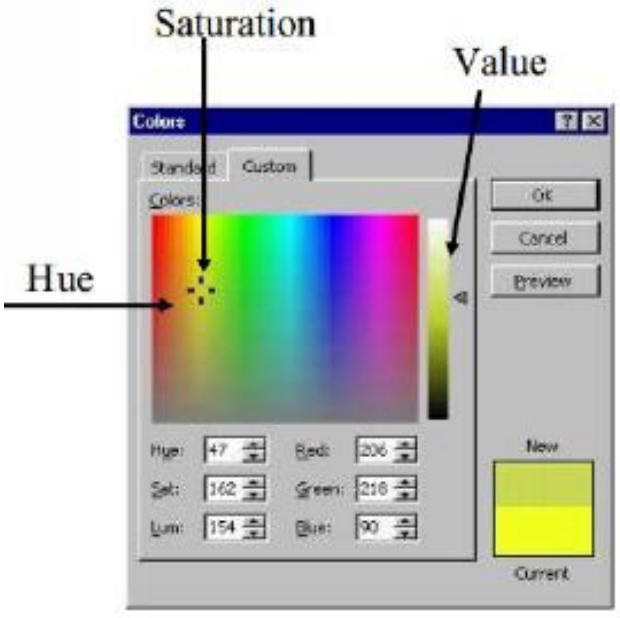
- H: هو التدرج اللوني. ويمثل بزاوية تدور حول محور المخروط أي عند الزاوية 0 يكون اللون أحمر.

- S: هو درجة إشباع اللون, والقيمة العظمى له تساوي 1 وهو يتمثل بالبعد عن مركز المخروط وكل ما اقترب منه يصبح اللون اقرب إلى تدرجات الرمادي وكل ما ابتعد عن مركز المخروط أصبح اللون مشبع أكثر وله لون فاقع اكثر.



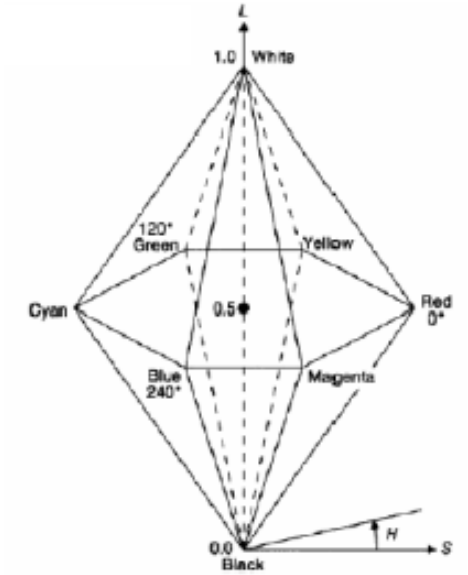
- V: وهي قيمة اللون أو السطوع له, إذا اقتربت قيمة السطوع نحو قاعدة المخروط أصبح اللون اغمق, وإذا كانت على القاعدة تماماً يكون اللون أسود نقي, القيمة العظمى له تعني أن اللون هو الأسطع,

التحويل من نظام RGB إلى هذا التنسيق هو تحويل غير خطي.

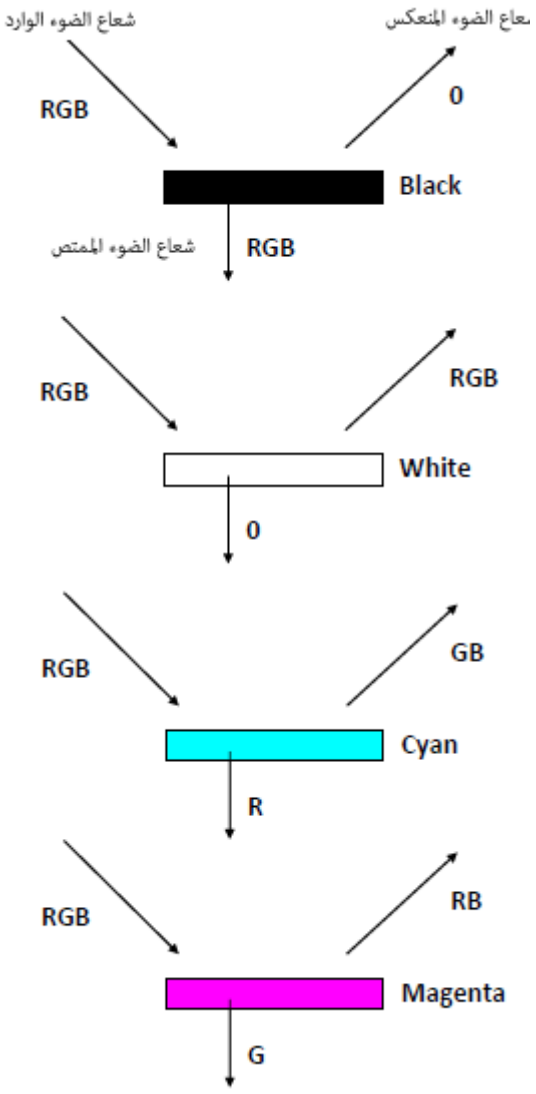


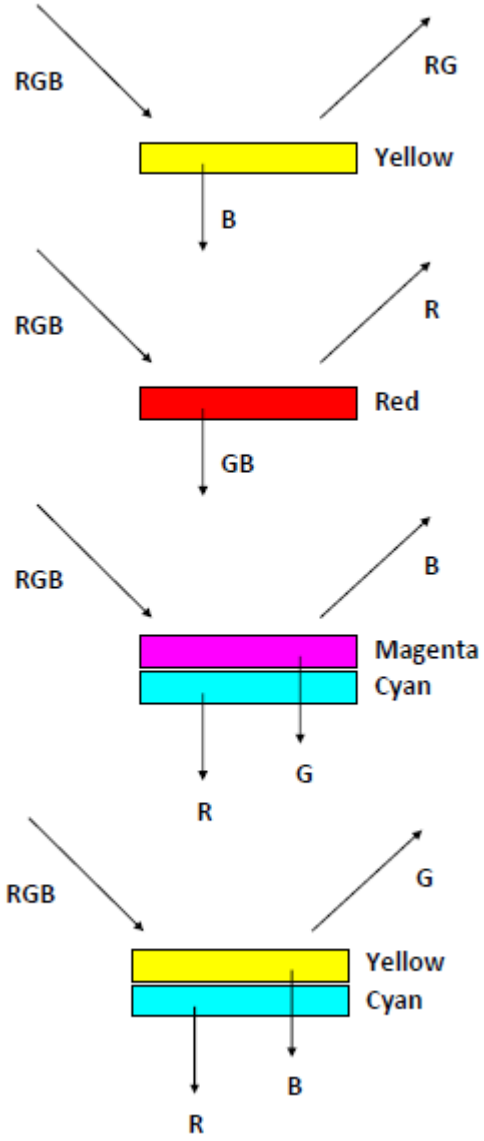
تنسيق الألوان HLS:

- يمثل بمخروط ثنائي.
- الاشباع الاعظم هو S=1
. L=0.5%
- عند L = 0% اللون هو اسود.
- عند L = 100% الون هو ابيض.



أمثلة لإنعكاس الضو وظهور الألوان:





الألوان

(في الفيديو)

م.عمر سليمان

29/02/2017

RBTCC#

نظم وسائط متعددة

أنماط الألوان في الفيديو:

- YIQ.
- YUV.
- YC_bC_r.

أنواع الألوان في الفيديو:

- لنظام الـ RGB عند العرض على الشاشات تظهر الألوان على شكل القيم الأساسية (أحمر – أخضر - أزرق).
- التلفاز يستعمل إشارة مركبة للعرض.
- الفائدة من نظام الإنارة والتصبغ أو التلون مثل نظام YUV ونظام YIQ هي أنه مازال يدعم اجهزة التلفاز ذات العرض باللون الأبيض والأسود.

مثال :

نظام الترميز NTSC يشكل إشارة مركبة للفيديو و تدعى أيضاً YIQ.

- NTSC ترميز يستخدم لبث اشارة التلفاز في شمال اميركا واليابان.
- وهذا الترميز كان مستعملاً أيضاً في شرائط الفيديو VHS في هذه البلدان منذ أن بدأ نظام الـ YIQ .

نظام الألوان YIQ:

- Y:
  - في نفسها الموجودة داخل الفضاء اللوني الممثل بـ XYZ.
  - معلومات الإضاءة (الإنارة) تكون موجودة ضمن هذه المركبة.
  - اجهزة التلفاز التي تعرض بالأبيض والأسود تستخدم إشارة Y فقط.

- I , Q:
  - المركبات هذه تحمل الألوان وترميزاتها.
  - I تحمل اللونين البرتقالي والأزرق السماوي.
  - Q تحمل اللونين الأخضر والأرجواني او البنفسجي.
- لكن الإنارة والتلوّن هي قيم موجودة على إشارات رقمية متعددة.

تحويل من RGB إلى YIQ:

- التحويل بين النظامين هو تحويل خطي.
- التحويل من نمط RGB إلى نمط YIQ يتم من خلال علاقة التحويل التالية:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.595879 & -0.274133 & -0.321746 \\ 0.211205 & -0.523083 & 0.311878 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- ويوجد تحويل آخر يستخدم من قِبل محولات الترميز NTSC:

(أي عند التحويل من YIQ إلى RGB عن طريق ترميز NTSC)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.107 & 1.704 \end{bmatrix} * \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

مثال لعرض طريقة فصل الألوان في نظام الألوان YIQ:


Original Color Image


Y Image


I Image


Q Image

نظام الألوان YUV:

- في أوروبا نظام الألوان YUV كان يستخدم في انواع الإتصالات PAL و SECAM والبث التلفزيوني في هذه المنطقة.
- ويستخدم في الخوارزميات الأساسية لضغط الفيديو مثل MPEG-2, ويستخدم أيضاً في نظام التلفاز الديجيتال والـ DVD.
- المركبات لهذا النظام : Y للإنارة وجمع الألوان و U و V للتصبغ أو التلون وتفريق الألوان.
- وجدت هذه الإشارات وبنيت على أساس نظام RGB.
- جميع الألوان دُمِجت ووضعت قيمتها داخل المركبة Y.
- الإشارة U قيمتها وجدت من طرح قيمة المركبة Y من اللون الأزرق في الصورة الأساسية.
- الإشارة V قيمتها وجدت من طرح قيمة المركبة Y من اللون الأحمر في الصورة الأساسية.
- قيم هذا النظام يمكن تحويلها بكل سهولة.

التحويل بين الRGB و YUV وبالعكس:

- طريقة إستخراج الـ YUV من ال RGB:

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

$$U = -0.299R - 0.587G + 0.886B$$

$$V = 0.701R - 0.587G - 0.114B$$

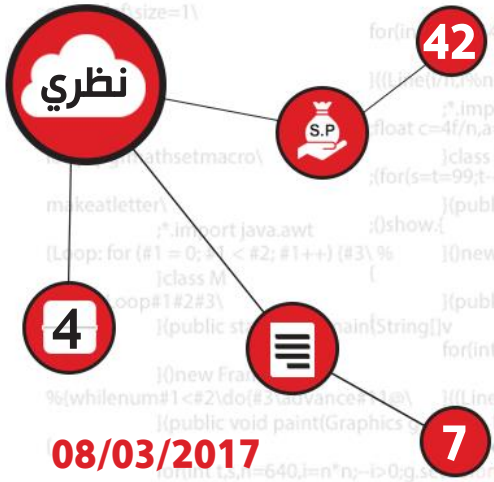
- التحويل من RGB إلى YUV بشكل تام:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.5287 & -0.114 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

نظام الألوان YC_bC_r:

- هو النظام العالمي الأساسي للفيديو الرقمي و المعروف أيضاً بـ Rec. 601 .
- يستخدم في ضغط الصور وفي ضغط الفيديو من صيغ JPEG و MPEG.
- عند الضغط وفك الضغط فإننا نفقد بعض المعلومات الطيفية للألوان ولكن هذا لا يوثر على دقة المحتوى
- التحويل يتم من خلال المعادلات التالية:

$$C_b = ((B - Y)/2) + 0.5$$
$$C_r = ((R - Y)/1.6) + 0.5$$



أساس الصورة الرقمية

م.عمر سليمان

08/03/2017

أخذ العينات: الدقة وابعاد البكسلات

- مرحلة تحويل الصورة إلى صورة رقمية تسمى مرحلة أخذ العينات(Sampling).
- اكثر ما يهمنا بالصور عند إلتقاطها هو مقدار عدد العينات المسجلة فيها والمحفوظة بشكل صحيح في مرحلة أخذ العينات.
- مقدار التفاصيل التي تأخذها الكاميرا عند الإلتقاط تسمى الدقة (Resolution).

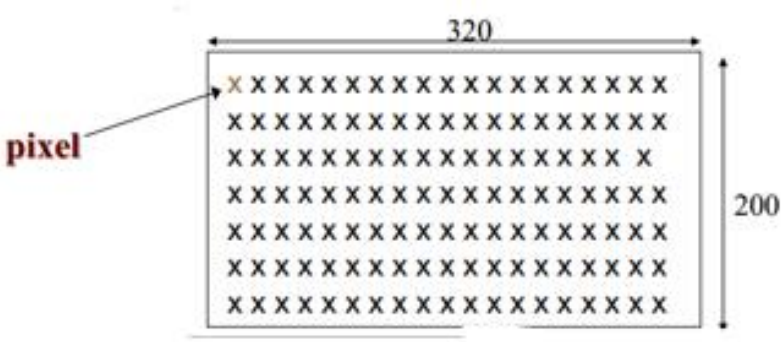
مقدار اخذ العينات:

- عدد المرات التي تم اخذ القياسات للعينات يحدد مستوى التفاصيل.
- عدد العينات التي نستعملها لعرض وخلق الصورة يسمى الدقة.
- زيادة عدد أخذ العينات يزيد حجم الصورة.
- الدقة الأعلى أي عدد العينات الأكبر دائما ينتج لنا صورة ذات جودة والوان أفضل.
- العينات: هي مصفوفة من الأرقام التي تمثل القيم للألوان.
- Bitmap: في مصفوفة ثنائية تحتوي التفاصيل عن كل بكسل داخل الصورة.
- الدقة تقاس بالبكسل.
- كلما زاد عدد البكسلات التي تلتقطها الكاميرا يزداد حجم الصورة دون الحصول على تفاصيل غير واضحة ضمنها.
- البكسل: هو أصغر عنصر داخل الصورة.
- يحتوي البكسل على قيم الألوان والأبعاد.

أخذ العينات ضمن الصورة:

1. إذا تم أخذ الصورة بعينة واحدة تكون الصورة بالأبيض والأسود وتدرجات الرمادي.
2. إذا تم أخذ الصورة بثلاث عينات فتتكون الصورة بالألوان الأساسية (أحمر-أخضر-أزرق).

- إذا تم أخذ الصورة بأربع عينات فهي تتكون من الألوان الأساسية ومن مركبة Alpha او ما يسمى بالشفافية.



أبعاد البكسلات:

الصورة تمثل بمصفوفة ثنائية تحوي قيم العينات وتسمى عناصر هذه المصفوفة بالبكسلات او العينات.

مثال:

صورة ذات ابعاد 200×320 تملك 320 بكسل أفقياً و 200 بكسل عامودياً

- **الدقة = العرض × الأرتفاع.**
- أبعاد الصورة هي نسبة العرض إلى الأرتفاع.
  - عادةً 4:3 وهي صورة عرضية.

الدقة في الكاميرا الرقمية

يوجد نوعين للدقة في الكاميرات:

1. الدقة البصرية: في الدقة التي تتولد من الحساسات داخل الكاميرا و تعتمد على عدد الحساسات ومدى قوتها.
2. الدقة الرقمية: لا تعتمد على ما هو فيزيائي في الكاميرا:
   - تعتمد على قوة المعالجة داخل الكاميرا وإضافة بكسلات للصورة الملتقطة.
   - البكسلات المضافة للصورة تكون بكسلات موجودة سابقاً لكن معدلة.

- الكاميرات الرقمية عادة يتم عرضها و تمييزها بعدد الميغا بيكسل.
- العدد الإجمالي للبكسلات في الصورة الرقمية يمكن حسابه من خلال صرب عدد البكسلات في العرض بعدد بكسلات الطول.

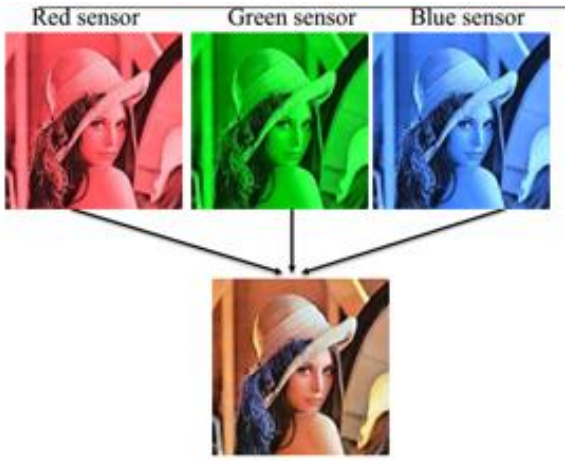
مثال:

1 600 * 1 200 = 1 920 000 pixels

One mega pixel = 1 000 000 pixels

1 920 000 pixels = 1.92 mega pixel

يتم تقريب عدد البكسلات عادة إلى اقرب رقم صحيح فنقول أن الكاميرا هي 2 mega pixel وليس 1.92 mega pixel



مجزء الأشعة:

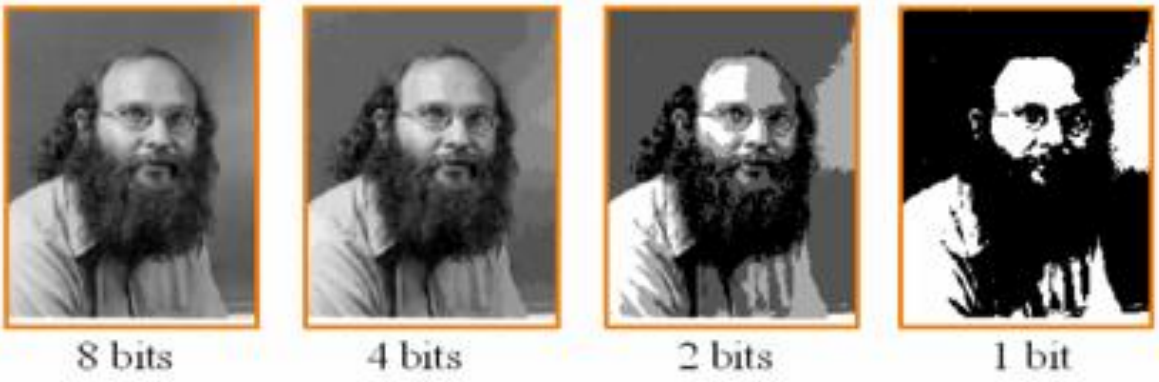
الصور الملونة عادةً تكون مقسمة إلى الألوان الساسية فيها وتلتقطها الكاميرا على ثلاث حساسات, حساس يميز اللوان الأحمر وآخر للأخضر وآخر للأزرق.

التقنيات المستخدمة لتخفيض حجم الصورة :

- تقليل مقدار أخذ العينات.
- تقليل العمق اللوني.
- الضغط.

العمق اللوني

- هو عدد البكسلات اللازمة لتمثيل البكسل الواحد.
- كل ما زاد عدد البتات في البكسل الواحد زادت نوعية ألوانه ودرجاته. ويزداد أيضا حجم الصورة الملتقطة.



الصور بدون ألوان:

Monochrome/bitmap Image (1 bit Images)

- كل بكسل يحتاج بت واحد لتخزينه
- الصورة غير ملونة
- تستخدم مع الرسوم البسيطة او التي تحتوي نصوص
- مثال : صورة ابعادها 480*640 تحتاج للتخزين

640*480=307200 pixel

307200 *1 bit=307200 bit

307200 /8=38400 byte=38.4 KB (1 KB=1000 byte)

Gray Scale Image (4 bit)

- كل بكسل يحتاج لـ 4 بتات لتخزينه
- الصورة تحوي تتدرج رمادي
- كل بكسل قيمته تتراوح بين 0 و 15
- 4 bit → 2^4 → 16 grey shades
- مثال : صورة ابعادها 480*640 تحتاج للتخزين

640*480=307200 pixel
307200 *4 bit=1228800 bit
1228800 /8=153600 byte=153.6 KB
(1 KB=1000 byte)

Gray Scale Image (8 bit)

- كل بكسل يحتاج لـ 8 بتات لتخزينه
- الصورة تحوي تتدرج رمادي
- كل بكسل قيمته تتراوح بين 0 و 255
- 8 bit → 2^8 → 256 grey shades
- مثال : صورة ابعادها 480*640 تحتاج للتخزين

640*480=307200 pixel
307200 *8 bit=2457600 bit
2457600 /8=307200 byte=307.2 KB
(1 KB=1000 byte)

الصور الملونة:

- 8 bit color images: تستخدم look up table
- 16 bit color images: اما ان يهمل البت في اقصى اليسار او ان يتم حجز 5 بتتات للأزرق و 5 بتات للاحمر و6 بتات للأخضر
- 24 bit color images: يتم استخدام 8 بت لكل مركبة لونية ويكون عدد الالوان المحتملة هو 16777216=256*256*256 لون
- 24 bit color images (truth image): يتم تخزين كل بكسل فعليا على 32 بت بدلا من 24 بت حيث يتم استخدام 24 بت للألوان و8 بت الاضافي للشفافية

مثال:

اذا كان لدينا صورة ابعادها 480*640 بكسل احسب حجم الصورة بالميجا بكسل في حال 8 bit color , 24 bit color

الحل:

الحالة الاولى

640*480=307200 pixel * 8 bit =2457600bit/8=307200 byte=307.2 mega pixel

الحالة الثانية

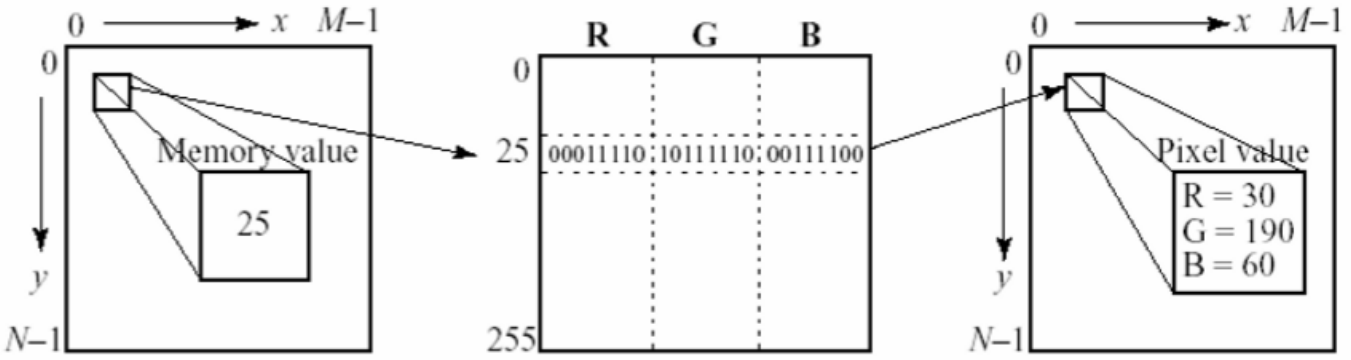
640*480=307200 pixel * 24 bit =7372800bit/8=921600 byte=921.6 mega pixel

8-bit Color Images

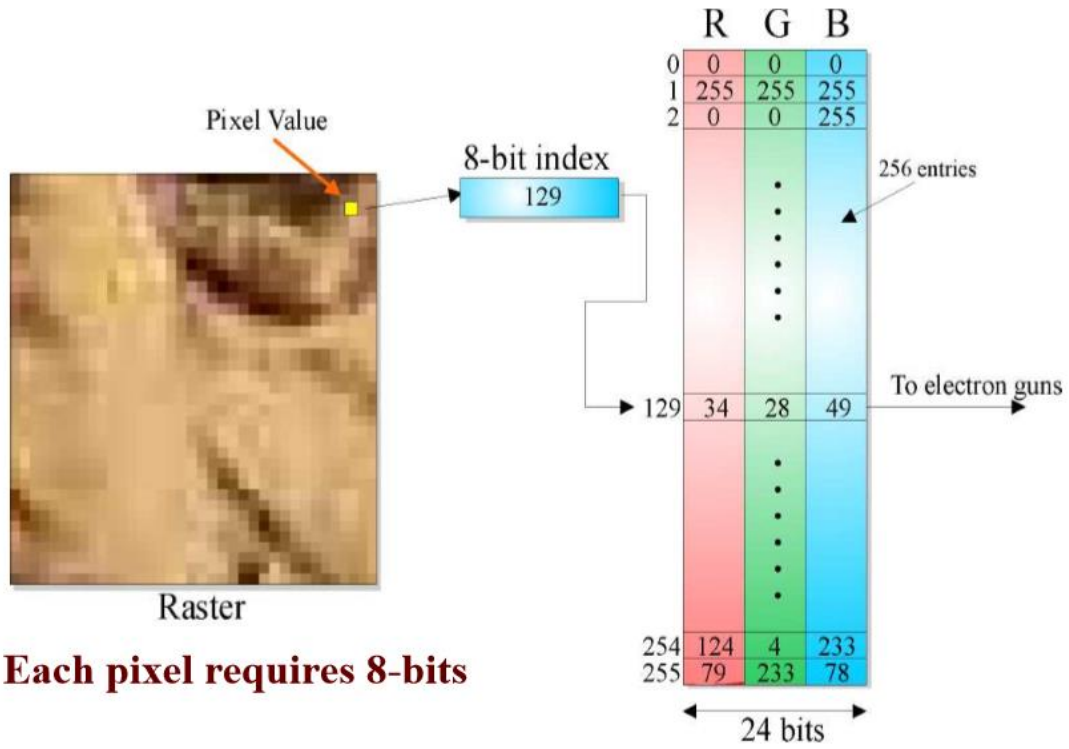
24-bit Color Images

Color Look-up Table (LUT) Indexed Color

- يتم في بعض الاحيان استخدام جدول لتخزين الالوان يدعى look-up table
- وهو عبارة عن جدول بعرض 24 بت وبطول 256 سطر
- كل سطر يعبر عن احد الالوان حيث كل 8 بت تمثل مركبة لونية (احمر اخضر ازرق)
- قيمة البكسل في الصورة لا تمثل اللون الفعلي وانما هي مجرد مؤشر إلى احد اسطر هذا الجدول (ما سيتم عرضه على الشاشة هو السطر المقابل للقيمة التي يحملها البكسل وليس قيمة البكسل مباشرة)



Color LUT for 8-bit color images.



الدقة وإعادة التحجيم

نميز نوعين لقياس الدقة في الأجهزة:

1. PPI: وتعني Pixel Per Inch أي مقدار البكسلات في الإنش الواحد وتستخدم هذه الطريقة للحساب في الصور الرقمية ويؤثر هذا العدد على دقة الصورة ويمكن الإستفادة منه في حساب الحجم الفعلي للصورة عند طباعتها.
2. DPI: وتعني Dot Per Inch أي مقدار عدد النقاط في الصورة ضمن الإنش الواحد ويستخدم لحساب دقة الصورة في الطابعات ولا يؤثر هذا العدد على حجم الطباعة للصورة بل يؤثر على دقة الصورة المطبوعة.

الدقة:

الحجم الفيزيائي = حجم البكسل \ دقة شاشة العرض.

مثال:

كاميرا تقوم بإلتقاط صورة بدقة 1200*1600 وطبعت هذه الصورة بإستخدام طابعة ذات دقة 300 dpi , أحسب الحجم الفيزيائي للصورة المطبوعة.

الحل:

1200 / 300 = 4 inch
1600 / 300 = 5.33 inch

مثال2:

صورة ذات حجم 6*4 إنش تم ألتقاطها عن طريق ماسح ضوئي بدقة 600 dpi.

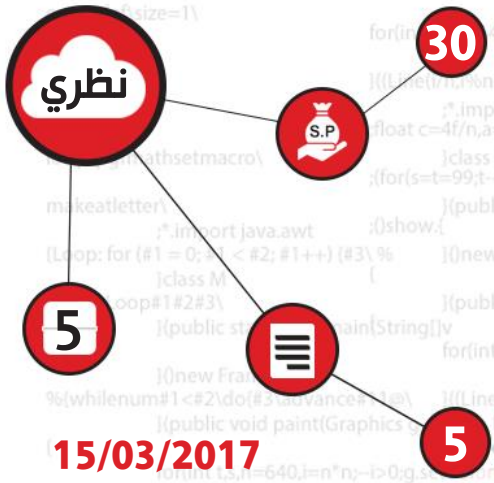
1. ما هي أبعاد الصورة الرقمية لها بالبكسلات؟
2. ما هو حجم الصورة عند عرضها على شاشة بدقة 72 PPI ؟

الحل:

6*600 = 3600 Pixel
4*600 = 2400 Pixel

3600 * 2400 Pixels

3600/72 = 50 inch
2400/72 = 33.3 inch
---
50 * 33.3 inches



أنواع الصور والصيغ

15/03/2017

م.عمر سليمان

لقد تعرفنا سابقاً على نوعين من الصور هي الـ Raster والـ Victor.

والـ Raster هو عبارة عن Bitmap أما الـ Victor في أشكال جاهزة تتكون من تعليمات يمكن توليدها عبر برامج مثل الـ AutoCAD و الـ OpenGL وغيرها من البرامج المتاحة.

سنتكلم عن:

- حفظ الصورة كـ Raster والصيغ واللواحق لها.
- أنماط الملفات الأكثر استخداماً وتوافقاً للويب.
- التخزين حسب كل نوع من الصيغ.
- والعتاد المطلوب لتشغيل كل صيغة.

ملاحظة: عند الإنتقال من صيغة لصيغة يوجد خصائص يمكن أن نفقدها.

الفئات لأنواع الصور تنقسم إلى:

- Victor.
- Bitmap أو Raster.
- Hyper: ويكون مزيج من النوعين السابقين.

للتذكرة:

الـ Victor يأخذ مساحة أقل عند التخزين لأنه عبارة عن تعليمات مسؤولة عن إظهار الصورة,

أما الـ Raster هي عبارة عن بكسلات وكل بكسل يحوي عدة خصائص لذلك تكون مساحته أكبر.

- عند إستخدام الـAutoCAD يمكننا تشكيل النوعين والتحيل بينهما.

معلومة عن الـ AutoCAD: إن هذا البرنامج يعتمد على مبدأ يسمى بالفاصلة العائمة, وهذا المبدأ يعني أن التطبيق مرتبط بما أنتجه أي أننا إذا اخرجنا صورة او ملف من هذا البرنامج فإن هذا الملف لايعمل سوى على هذا البرنامج.

الصور:

كل صورة يتم تحليلها بشكل معين ويبدأ التحليل من ما يسمى بالـ Signature او التوقيع او البصمة وهذا التوقيع يحدد نمط الملف او نوعه (JEPG - TIFF - GIF). ويلي هذا التقسيم ما يسمى بـ Roadmap او خريطة الموقع وضمنها يوجد الكثير من الأشياء التي توصف الصورة أي هل الصورة مضغوطة أم لا, نسبة الأخطاء بداخلها. بعدها يوجد ما يسمى بال Row Data وهي كل محتويات الصورة.

لواحق الصور: (PBM - PGM - PPM)

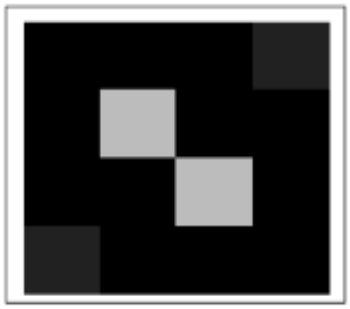
- **المميزات:**
  - مصممة بإمكانيات سهلة.
  - هي من مشتقات الـ Bitmap.
  - مكتوبة بالـ ASCII.
  - غير قابلة للضغط.
  - لا تتسع سوى لصورة واحدة بالملف.
  - ذات دعم قوي للمنصة الـ UNIX.
  - إمكانية قراءتها وكتابتها وهي ممتازة للصور التبادلية(**أي لتغيير لاحقتها**).
- **PBM:**
  - للصور من نوع Monochrome.
  - **مثال:**


```
# feep.pbm
  24 7
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0
  0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0
  0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0
  0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
  0 1 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

- **PGM:**
  - تقبل تدرجات الرمادي.
  - تتراوح القيمة اللونية لها بين الـ 0 و الـ 15.
  - **مثال:**

```
P2
# feep.pgm
24 7
15
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  3  3  3  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15 15 15 15  0
0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0 15  0
0  3  3  3  0  0  0  7  7  7  0  0  0 11 11 11  0  0  0 15 15 15 15  0
0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0  0  0
0  3  0  0  0  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

- **PPM:**
  - تعمل على تقسيم الصورة لبكسلات لتمثيلها.
  - **مثال:**



```
4  4
15
 0   0   0     0   0   0     0   0   0     0   0  15
 0   0   0     0  15   0     0   0   0     0   0   0
 0   0   0     0   0   0     0  15   0     0   0   0
 0   0  15     0   0   0     0   0   0     0   0   0
```

اللاحقة GIF:

- هي إختصار لـ Graphics Interchange Format, ويتم دعم هذه اللاحقة من قبل شركة CompuServe.
- هي واحدة من أشهر اللاحقات على الأطلاق:
  - هي واحدة من بين اللاحقات الأكثر دعماً على متصفحات الويب.
- مصممة بشكل رئيسي من صور الـ Raster المتغيرة(**أي تدعم التبادل**).
- هذه اللاحقة تدعم الشفافية.
- تقبل الضغط وبعد الضغط هي تكون الأقل ضياعاً للمعلومات مقارنة بباقي اللاحقات.
- تعمل فقط على ما يسمى بالصور ذات الفهرسة اللونية Indexed Color Images:

- تدعم حتى 256 لون.

- تدعم النظام اللوني 24-bits وحجم الصورة يصل إلى 64k * 64k Pixel.
- يضيع كمية كبيرة من المعلومات في هذا النوع من الصور.
- يمكن تحويل الصورة من النظام اللوني 24-bit إلى 8-bit.

- يدعم كافة المنصات التشغيلية Unix ومعظم الحواسيب الشخصية.
- تدعم خاصية الستارة Interlacing: أي أن إظهار الصورة يتم مثل الستارة الأفقية أو على شكل أسطر وخصائص الظهور للأسطر تكون مسجلة في ال RoadMap, مثال للتوضيح:

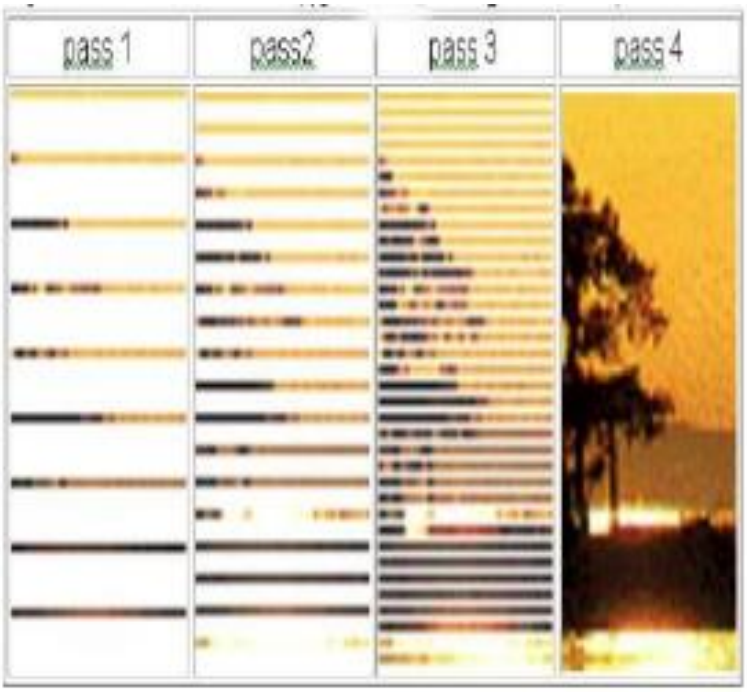


Image row	Pass 1	Pass 2	Pass 3	Pass 4	Result
0	*1a*				*1a*
1				*4a*	*4a*
2			*3a*		*3a*
3				*4b*	*4b*
4		*2a*			*2a*
5				*4c*	*4c*
6			*3b*		*3b*
7				*4d*	*4d*
8	*1b*				*1b*
9				*4e*	*4e*
10			*3c*		*3c*
11				*4f*	*4f*
12		*2b*			*2b*

يوجد للـ GIF إصدارين:

- GIF87a: هو الإصدار بالخصائص الأساسية.
- GIF89a: هو الإصدار الأحدث, يدعم التحريك في الصور بشكل بسيط, يدعم الشفافية.
  - دقتها ضعيفة.
  - تم تحسين الترويسة التي تقوم بحفظ البيانات عن الصورة.
  - لها أدوات مجانية كثيرة تقوم بصناعتها.
  - غير جيدة للملفات الكبيرة والملفات ذات المدة الطويلة.
- صور ال GIF بشكل عام تدعم خوارزمية الضغط LZW Lossless أي أنها بعض الضغط وفك الضغط تكون أقل ضياعاً للمعلومات.

اللاحقة JEPG:

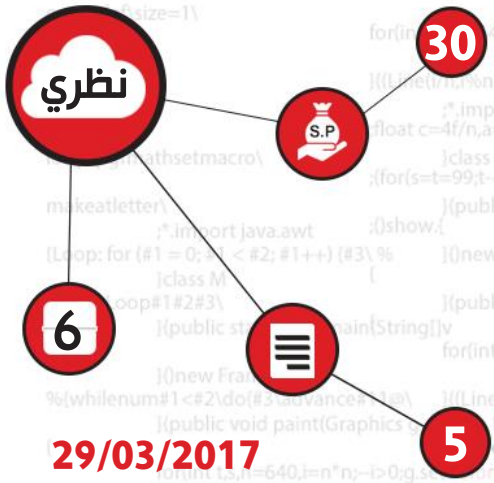
- تم تطويرها من قبل Joint Photographic Experts Group.
- تم تصميمها للصور ذات الألوان عالية الدقة.
  - يتم تمثيلها بـ 24-bit.
- تدعم الويب.
- نظامها اللوني الأساسي RGB, CMYK.
- متكاملة مع كافة منصات التشغيل.
- تدعم الضغط وتقبل الضغط بضياع أقل للمعلومات.
  - تقبل الضغط بنسبة تصل إلى 30:1.
  - إن الضغط وفك الضغط يتم بصورة بطيئة بسبب حجم الملف.

اللاحقة PNG:

- PNG هي إختصار لـPortable Network Graphics.
- الملفات بهذه الصيغة تدعم الويب, وتم وضعها لتستبدل الملفات بصيغة GIF.
- تدعم الضغط قليل الضياع.
- تعطي صور أعلى دقة من الـ GIF, لكن لاتدعم الحركة في الصورة مثلها.

الصيغ اللونية لها:

- 1, 2, 4, 8, 16 bit لكل بكسل - لتدرجات الرمادي.
- 1, 2, 4, 8 bit لكل بكسل – ضمن التمثيل بالفهرسة اللونية.
- 8, 16 bit لكل قناة – ضمن الفضاء اللوني RGB.
- ويسمح بإستخدام عنصر الشفافية لكل قناة Alpha (تصبح الصيغة 64bit per pixel).



أنواع الصور والصيغ 2

م.عمر سليمان

29/03/2017

RBTCC#

نظم وسائط متعددة

ملاحظة هامة للمحاضرة السابقة:

- نوع الضغط في ملفات الـ GIF87a هو LZW lossless.
- نوع الضغط في ملفات الـ PNG هو LZ77 Compression.

اللاحقة TIFF:

- تدعم الصيغ اللونية التالية: الأبيض والأسود – تدرجات الرمادي – RGB - CMYK - YCbCr- CIELab.
- يوجد لها خمس أنواع للضغط:
  - RLE (Run Length Encoding).
  - LZW (Lempel-Ziv-Welch).
  - JEPG.
  - ويمكن استخدامها بدون ضغط.
- يتضمن الملف الخاص بها صورة وحدة.
- تواجه مشكلة بالمحمولية أي عند نقلها من نظام لنظام قد تواجه مشاكل.

اللاحقة BMP:

- اتت كبديل عن عدة صيغ سابقة وتدعم الويندوز بشكل كبير.
- تدعم الصيغ اللونية التالية:
  - 1bit للأبيض والأسود.
  - 8bit لتدرجات الرمادي.
  - 16, 24, 32bit لنظام RGB.
  - 4, 8bit للتمثيل الجدولي للألوان Indexed Color او ما يسمى LookUp Table.
- تدعم الشفافية للصورة وتدعمها لكل قناة على حدى أيضاً.
- تدعم الضغط ضمن خوارزمية RLE.

اللاحقة PCX:

- قامت بتطويرها شركة Zsoft, وظهرت لدعم منصات الPC.
- وتكون هذه الصيغة غير فعالة في حالة الصور المسحوبة على الماسح الضوئي وحالة الفيديو.
- لا تدعم التحويل من النظام اللوني CMYK إلى نوع أخر أو بالعكس.
  ولا تدعم التحويل من النظام اللوني HSI.
- وتدعم 8bit لونية مع 256 لون.

اللاحقة EXIF:

- تعاني من مشكلة التاغات والتي هي أكثر من لاحقة TIFF.
- يمكن استخدامها بالطابعات وبمجال تصحيح الأخطاء للخوارزميات.
- جيدة للطباعة.

اللاحقة PAINT و PICT:

- ظهرت على أجهزة الMAC ثم بدأ دعمها من قبل ويندوز.

الملفات من نوع Metafile أو الهجين ونحن غير مطالبين به.

جداول لتوضيح المعلومات السابقة:

Bitmap Images

File Suffix	Our Abbreviation	File Type	Characteristics
.bmp	BMP	Windows bitmap	1 to 24-bit color depth, 32-bit if alpha channel is used. Can use lossless RLE or no compression. RGB or indexed color.
.gif	GIF	Graphics Interchange Format	Used on the web. Allows 256 RGB colors. Can be used for simple animations. Uses LZW compression. Originally proprietary to CompuServe.
.jpeg or *.jpg*	JPEG	Joint Photographic Experts Group	For continuous tone pictures. Lossy compression. Level of compression can be specified.
.png	PNG	Portable Network Graphics	Designed as an alternative to *.gif* files. Compressed with lossless method. 1 to 64-bit color with transparency channel.
.psd	PSD	Adobe Photoshop	Supports a variety of color models and bit depths. Saves image layers created in photographic editing.
.psp	PSP	Corel Paint Shop Pro	Similar to *.psd*.
.raw		Photoshop	Uncompressed raw file. Could be black and white, grayscale, or RGB color.
.tif or *.tiff*	TIFF	Tagged Image File Format	Often used for traditional print graphics. Can be compressed with lossy or lossless methods, including RLE, JPEG, and LZW. Comes in many varieties.

Vector Graphics

File Suffix	Our Abbreviation	File Type	Characteristics
.ai	AI	Adobe Illustrator	Proprietary vector format.
.swf	SWF	Shockwave Flash	Proprietary vector format; can contain stills, animations, video, and sound.
.cdr	CDR	Corel Draw	Proprietary vector format.
.dxf	DXF	AutoCAD ASCII Drawing Interchange Format	ASCII text stores vector data.

حجوم الملفات للصور حسب الصيغ:

mage Size	TIFF (uncompressed)	JPEG (high quality)	JPEG (medium quality)
640x480	1.0 MB	300 KB	90 KB
800x600	1.5 MB	500 KB	130 KB
1024x768	2.5 MB	800 KB	200 KB
1600x1200	6.0 MB	1.7 MB	420 KB

الفهرسة

Color Quantization: Indexed Color

هي مرحلة تمر بها الصورة عند تحويلها من الحالة الورقية للحالة الرقمية.

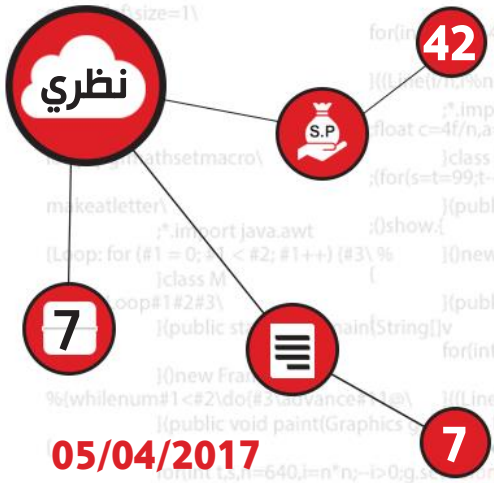
من خلالها يتم تقريب اللون الموجود بالصورة الورقية إلى الوان مشابهة بالحالة الرقمية.

الألوان التي تم التقريب إليها هي الوان موجودة في جدول وهي مفهرسة, اي يصبح كل بكسل من الصورة يؤشر إلى مكان معين ضمن الفهرس وضمن الفهرس هذا المكان يكون له لون وعند العرض تظهر الألوان حسب المؤشرات. **مثال:**

بهذه الحالة قد قمنا بإختصار الكثير من المساحة التي كنا سنحتاجها لتخزين كل لون داخل كل بكسل على حدى.

مراحل عملية الفهرسة:

1- يتم تحديد المجال الفعلي للألوان في الصورة.
   a. إذا كانت الصورة ضمن المجال 24bit فإنه يوجد لنا:
      2^{24} = 16,777,216 احتمال لوني.
2- يتم بعدها وضع القيمة المراد تحويل الصورة ضمنها مثلا إذا أردنا أن نجعل الصورة 8bit عند التحويل لرقمي.
   a. عند جعل الصورة 8bit فالإحتمالات اللونية 2^8 = 256.
3- بعد أن حددنا الألوان المستخدمة في الصورة نضعها ضمن الجدول ونجعل كل بكسل يشير إلى اللون المخصص له وبذلك نكون قد قللنا من مساحة الصورة وقللنا العمق اللوني لها.



التدرج اللوني

05/04/2017
م.عمر سليمان

نظم وسائط متعددة

عند الحاجة لعرض صورة على شاشة لاتدعم كافة الوان الصورة يوجد عدة حلول منها:

- جدول الألوان الموحد (Uniform Color Table).
- الالوان الشائعة Popularity.
- Median cut.
- Octree.
- Dithering.
- Floyd Steinberg.

الصورة التي نريد المناقشة فيها هي:

وهية صورة Raster ذات ألوان واضحة ودقة عالية.

Uniform Color Quantization method

الميزات والمساوئ:

تعمل على مبدأ التقريب اللوني يعني إذا كان اللون رمادي يتم تقريبه للأسود وإذا كان دون رمادي فإننا نضعه بالأبيض.

نأخذ مكعب الالوان الموجود في الصورة كل بكسل على 24 بت نقسم مكعب الالوان الى 256 مكعب جزئي تسمى منطقة، نقرا البكسل من الصورة ونضعها في المنطقة الخاصة به

بعد قراءة كافة البكسلات في الصورة ستتوضع في مناطقها 256 منطقة
و256 لون ستكتب في جدول الالوان lookup لتمثل ألوان الصورة
إن هذه الطريقة:

- سهلة بسيطة سريعة لا تعبر بشكل كافي عن الالوان الموجودة بالصورة.
- النتيجة سيئة في الصورة.
- لا تهتم بالألوان الموجود في الصورة تقرا بكسلات وتضعها وتوزعها في LookUp .

طريقة التقسيم للتدرجات:

- نحضر مكعب الالوان RGB ثم نقسمه لـ 256 منطقة.
التقسيمات تكون مكعبات متساوية أو للازرق تقسيمات اقل مثلا.
- يكون لكل مكعب جزئي الثلاث قيم بشكل موحد.
- وذلك نكون قد حولنا من 16 مليون لون إلى 256 لون.
- وعندما نريد لون نختاره من التقسيمات الجديدة.

8 bits per pixel in this image

المساوئ:

نتائجها ليست جيدة – لان 256 لون تم اختيارها من دون دراسة مهما كانت الصورة.
واضح ضعف الخوارزمية بالتدرج والسبب ان الالوان في الجدول لاتمثل الصورة لانها ثابتة ليس بالضرورة ان تكون جيدة

Popularity Quantization method

بدلا من 256 منطقة سنقسم كل محور الى 64 تدريجة (**زدنا المناطق وسناخذ الالوان الاكثر تعبيرا**).
نقرا الصورة ببكسلاتها ونوزعها على 262 الف منطقة (64*64*64).
في هذ التنسيق نريد المناطق التي تحوي عدد بكسلات اكثر الالوان الاكثر شيوعاً لنعبر عن الالوان الاكثر شيوعاً

هذه الطريقة سهلة بسيطة تحتاج لوقت للتنفيذ حجوم ذواكر، الاهمية تعطي الوان افضل واكثر شيوعاً

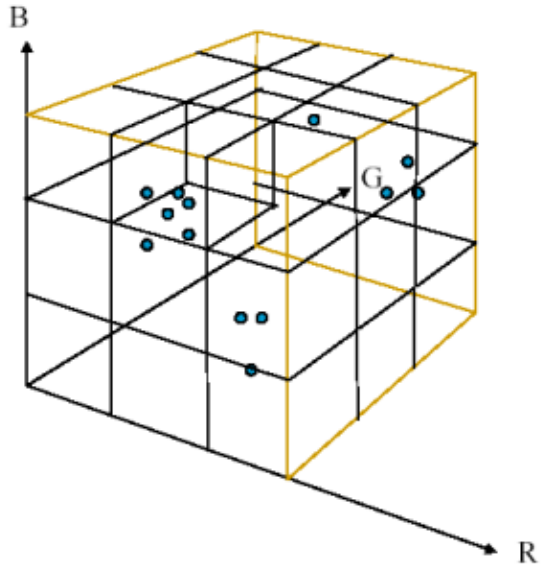
السلبيات:

اللون الاكثر شيوعاً تم التعبير عنه بشكل افضل.
الالوان الاقل شيوعاً في الصورة سيظهر بلون مختلف كما سنرى في الصورة. (**قد تكون الالوان ذات اهمية في الصورة وهذا يعتبر عيب في هذا التنسيق**).

طريقة التقسيم للتدرجات:

- مكعب الالوان RGB نقسمه الى 16 مليون مكعب 262 الف مكعب صغير.
- نفتح ملف الصورة الاصلية ونقرأ بكسلاتها كل بكسل على حدى ونقوم بتوزيعهم على احد المكعبات السابقة يتم توزيع البكسلات على المناطق، مثلا منطقة فيها 10 بكسلات ومنطقة 1000 بكسل نأخذ الالف لأنها تمثل الصورة افضل
- نرتب المناطق تنازلي للاقل وناخذ اول 256
- طريقة طويلة تاخذ ذواكر أكثر وتاخذ الألوان الاكثر شيوعاً.

8 bit image, so the most popular 256 colors



ملاحظة: اللون الأزرق للكرة الموجودة بسبب قلت تواجده بالصورة فإنه ظهر باللون الرمادي.

Median Cut method

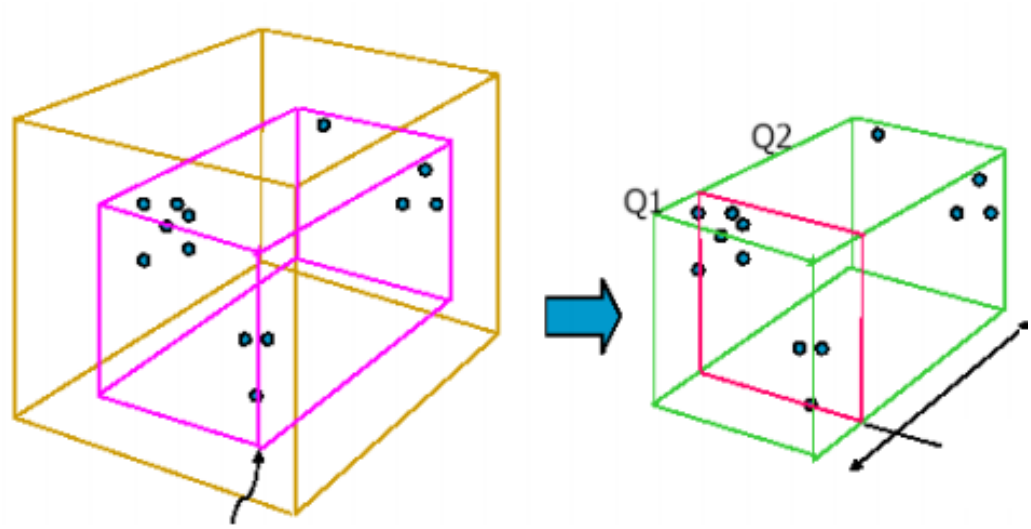
تعتمد على توزيع الالوان الاصلية وتحاول البحث على اصغر مكعب يحوي كافة الالوان في الصورة وتفرزها وفق القيم الاكبر. بدلا من المكعب كامل نلاحظ بالصورة ان الالوان محصورة داخل المكعب.

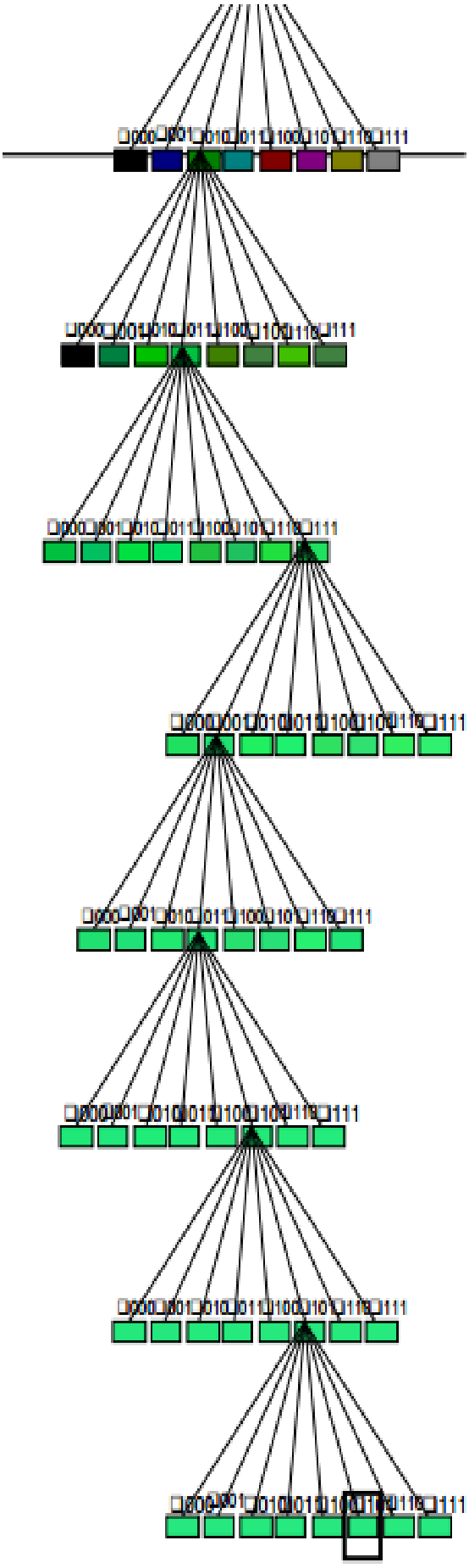
نوزع البكسلات على مكعب الالوان نفرز الالوان الناتجة حسب المحور الأطول RGB .(**من الأكبر للأصغر**)

ثم نقسم المتوازي حسب محور الأطول الذي ينتج نتيجة التقسيم اول مرة لمنطقتين (**مثل خوارزمية البحث الثنائي**).

المفاتيح متساوية بالقسمين، تقريبا، عدد البكسلات على اليمين = على اليسار ثم نكرر ناخد منطقة ونقسمها نكرر حتى نصل الى 256 (**مثل التقسيم الشجري**).

حاولنا نضمن ان كل منطقة فيها نفس عدد البكسلات تم التوفيق بين كل الالوان وبالنتيجة كل منطقة فيها 520 بكسل لكل مركبة نجمع ونقسم على العدد لنحصل على اللون





Octree Quantization method

نقسم المكعب الى 8 عقدة يتم ترقيمها بالأعداد بالنظام العد الثنائي، ثم نقسم العقدة الابن الى 8 وهكذا نكرر (شجرة معكوسة) حتى اخر عقدة. نسميها الورقة،

نقرا بكسل من ملف الصورة ، 8 لكل مركبة لونية،ومنها واحد لكل لون. 1 لون 0 لون 1 لون.

مثلا 101 سيكون في مكعب جزئي

نقرا 1 1 1 نضعها بالابن

001 بالحفيد وهكذا سيكون مستقر في مكعب جزئي ضمن المكعب اللوني

- **بمعنى آخر عند بناء كل إبن يتم إعادة ترقيمه من البداية الصورة للتوضيح**

بالاخير سيكون لدينا 256 ورقة بالشجرة المعكوسة، وعندما يزيد عدد الاوراق على 256 بواحد سنأتي لعقدتين متجاروتين لاب واحد، نجمع كل لونين منهما ونحسب المتوسط، ونضع بالعقدة الأب قيم الابنين.

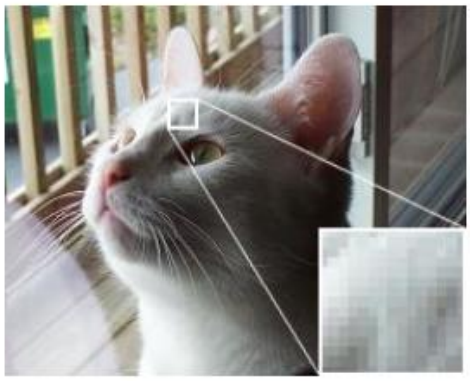
تسمى هذه العملية: reduction

عند اخذ اللون ووضعه في الجدول سنأخذ المفتاح ونربطه مع ملف الصورة.

Dithering

الاستفادة من خاصية العين البشرية في مزج الالوان، العين ستقوم بمزج الالوان لرؤية الوان غير موجودة على الشاشة او الصورة هي الفكرة الاساسية في ال dithering شكل من اشكال ال quantization

مثال:



Bi-level Thresholding

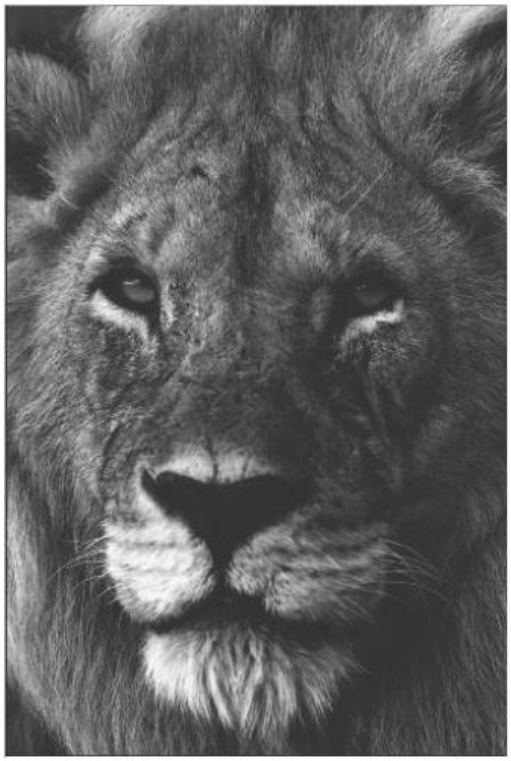
لتحويل صورة رمادية الى صورة ابيض واسود

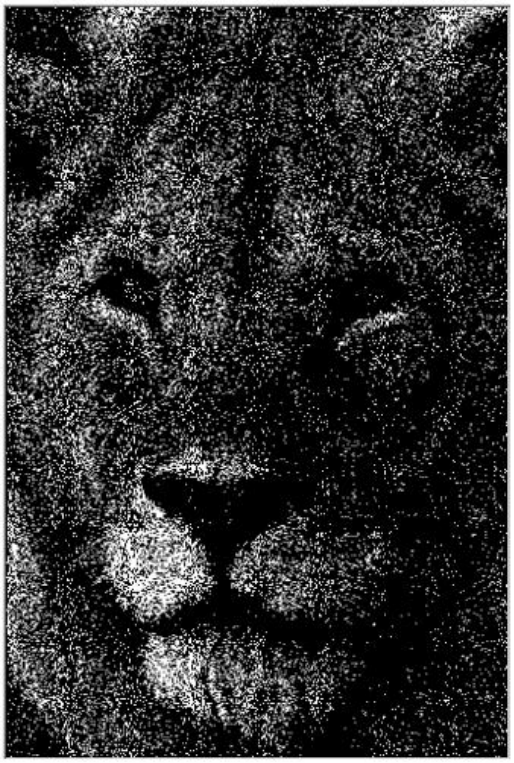
نقوم بتحديد عتبة قيمة رقمية معينة 126 كل بكسل اكبر منه ابيض واقل منه يكون اسود

يوجد تقانات لاتلجئ لحد عتبة ثابت وانما تلجئ لهيستو غرام واستخراج حد العتبة منه

الهيستوغرام: منحني يمثل عدد البكسلات بالصورة من أجل كثافة لونية معينة، من اجل بكسل كثافته الضوئية 1 مثلا عند 7 مرات وهكذا

Random Dithering (noise Dithering)

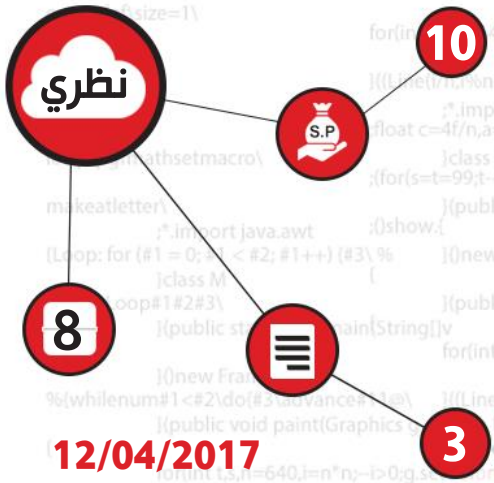




Random Dithering (noise Dithering)

Original image

Random Dithering



ضغط الصور والملفات

م.عمر سليمان

12/04/2017

RBS TCC#

نظم وسائط متعددة

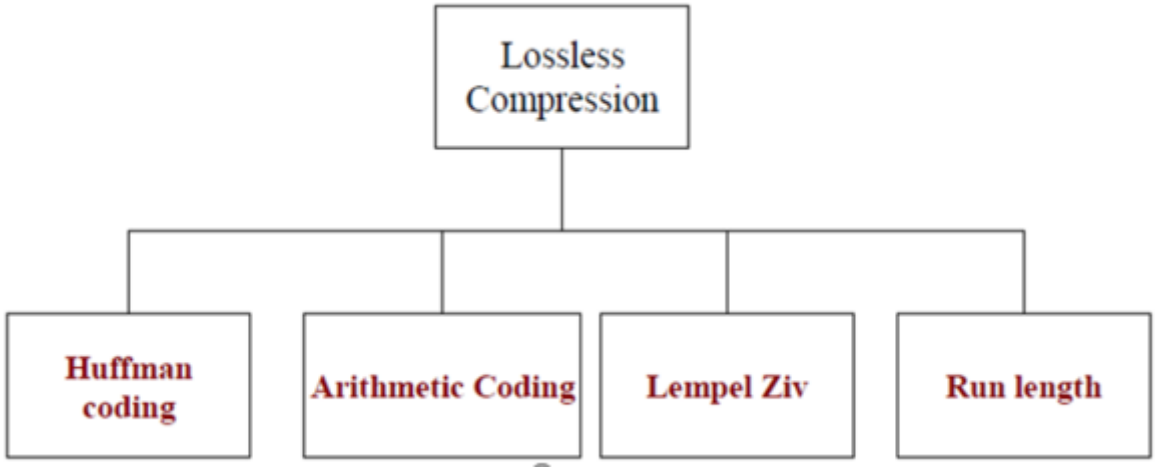
تطور تقنيات الوسائط المتعددة جعل حلم المكتبة الإلكترونية يتحول إلى حقيقة. الآن كل المكاتب والمتاحف والحكومات تقوم بتحويل الأرشيف لديها إلى ارشيف الكتروني, إن هذه الأرشفة سوف يكون حجمها هائل بالطبع ويجب أن يتم حفظ وتخزين هذه الأرشفة دون فقدان أي معلومات منها, إذاً ما الحل؟

الحل هو استخدام أساليب الضغط للتقليل من الحجوم المستخدمة.

في الصورة يمكن التضحية ببعض المعلومات من أجل التقليل من حجم الصورة,في الصوت يمكن التضحية بمعلومات أكثر لأن الأذن لا تميز التغيرات في الصوت بشكل جيد كما تميز العين تدرجات الألوان, أما في الملفات النصية فلا يمكن التضحية أبداً بالمعلومات الموجودة فيها.

يوجد نوعان من الضغط للملفات:

1. **Lossless Compression**: يتم فيه ضغط الملفات دون فقدان أي معلومات من الملف, من الخوارزميات المستخدمة في هذا النوع من الضغط مايلي:



2. **Lossy Compression**: يتم فيه ضغط الملفات لكن الخوارزميات المطبقة في هذا النوع من الضغط قد تسبب بفقدان بعض المعلومات.

ضغط الصور:

كما نعلم أي ملف مهما كان نوعه يتم تخزينه على شكل سلسلة من الأصفار والواحدات, إن فكرة الضغط هي الإستفادة من التكرار في هذه القيم. بفرض لدينا صورة من نوع 1bit image أي البيكسل فيها يمكن أن يأخذ قيمتين فقط. إذا كانت قيمة البكسل 0 أي اللون أبيض وإذا كان 1 يكون اللون أسود, يكون الملف مخزن على الشكل التالي:

000111111100000111100000

100011100000010111001111

.....

يمكننا ضغط الصورة السابقة بكتابة عدد مرات تكرار كل قيمة عوضا عن كتابة القيمة نفسها بعدد التكرارات يصبح شكل السطر الأول من الصورة بعد الضغط على الشكل التالي:

37545

نقوم بتخزين معلومات إضافية في السطر لكي نستطيع فك الضغط عندما نريد ذلك, مثلا نخزن أن السطر يبدأ بقيمة صفر لذلك أول رقم في الملف المضغوط يعبر عن عدد مرات تكرار القيمة 0 بينما القيمة التي تليها تعبر عن عدد مرات تكرار القيمة 1 وهكذا.

وضوحاً نجد أن الضغط بهذه الطريقة يكون مكلف جداً إذا كانت الصورة شطرنجية او لا تعبر عن تجانس طبيعي بالجوار, بينما يكون الضغط فعال وممتاز إذا كانت الصورة حقيقية ومتجانسة.

في الصورة الملونة RGB 24bit color احتمال التجانس قليل جداً لأن البيكسل الواحد له ثلاث قيم يتم تخزينها بشكل تسلسلي في الملف, لذا احتمال أن تكون قيمة بكسل ما متجانسة مع قيمة بكسل بجوارها يساوي ما يقارب 1/16 مليون لون بينما إذا تم تخزين الصورة بتسلسل آخر, مثلاً نقوم بتخزين قيم اللون الأحمر في كافة البكسلات بجانب بعضها وبعدها الأخضر وبعدها الأزرق فيصبح احتمال أن بتجانس بكسلين متجاورين أكبر بكثير من الإحتمال السابق حيث تصبح قيمته الآن 1/256 لون ويتم الضغط بنفس الطريقة السابقة.

ملاحظة:

يوجد خوارزميات لضغط الصور تقوم بحساب قيم الألوان للبكسلات الثمانية المجاورة لأي بكسل الصورة وتقرر إذا كانت هذه البكسلات شاذة عن مجاورتها فتقوم بحذفها أو تجانسه معهم وتنتقل إلى البكسل التالية وهكذا. لكن لهذا النوع من الخوارزميات طويل وبطيء بسبب كثرة العمليات التي تقوم بها ولكن نجد أنها فعالة في الصور ذات الأبعاد الصغيرة.

ضغط النصوص:

بفرض لدينا الجملة التالية باللغة العربية : "الجمهورية العربية السورية" ونريد تخزين هذه الجملة في جدول ال ASCII (بالحقيقة يتم تخزين الأحرف العربية في جدول الUnicode وليس الASCII لكن المثال من كل محرف يأخذ حجم تخزين 1Byte في جدول الASCII لذا يكون حجم تخزين الجملة السابقة مع إعتبار الفراغات محارف 10 + 8 + 7 = 25bytes).

لنضغط الجملة السابقة نعرف أحرف خاصة بهذه الجملة في جدول الASCII أي نبحث عن تركيبات الأحرف التي تتكرر كثيرا في الجملة ونقوم بتعريفها كمحرف واحد. في الجملة السابقة نجد أن التركيبين: (الـ - ـية) يتكرران ثلاث مرات فنقوم بتعريفهم كمحارف جديدة في جدول الASCII فنجد أن حجم تخزين الجملة نفسها أصبح: 8 + 6 + 5 = 19byte.

نسمي هذه الطريقة بالضغط الإحصائي لأننا نقوم بتطبيق خوارزميات تقوم بالبحث عن التكرارات داخل النص. يمكن أن يكون التكرار لتركيب أكثر من محرف أو يمكن أن يكون لكلمة كاملة.

خوارزمية RLE: Run Length Encoding:

الفكرة الأساسية في هذه الخوارزمية تشبه الخوارزمية التي تم شرحها في فقرة ضغط الصور.

وهي فكرة الإستفادة من التكرار في المعلومات. سنقوم بتطبيق الخوارزمية على النصوص أولاً ثم نطبقها على الصور بفرض لدي الجملة التالية التي أريد ضغطها باستخدام هذه الخوارزمية:

All is too well

بتطبيق الخوارزمية حيق نضع عدد التكرار قبل المحرف المكرر, تصبح الجملة على الشكل التالي:

A2l is t2o we2l

لكن تخزين الجملة بالشكل السابق سوف يسبب مشاكل عند فك الضغط وقراءتها من جديد لأن الأرقام تعتبر محارف. إذاً كيف سأميز إذا كان هذا الرقم من أصل الجملة أم هو عدد مرات تكرار المحرف الذي يلي الرقم؟

حلت الخوارزمية هذه المشكلة بشكل جزئي بوضع المحرف "@" قبل الرقم الذي يمثل عدد مرات النكرار أي يتم تخزين الجملة السابقة على الشكل التالي:

A@2l is t@2o we@2l

مشاكل هذه الخوارزمية في ضغط النصوص:

- إن المحرف "@" قد يكون متواجد في النص أيضاً.
- يتم تطبيق هذه الخوارزمية عند وجود ثلاث تكرارات أو أكثر من المحرف نفسه.
  لكن في النصوص احتمال ورود ثلاثة محارف متشابهة قليل جداً لذا تعتبر الخوارزمية غير فعالة مع النصوص.
- الرقم الذي يعبر عن التكرارات له حد معين وهو 255 أي إذا كان لدي جملة فيها 300 تكرار من الحرف a سيتم تخزين هذه التكرارات على مرحلين على الشكل التالي:

255a 45a

وجدنا أن هذه الخوارزمية غير فعالة في ضغط النصوص لكنها تستخدم كثيراً في ضغط الصور وخصوصاً الصور التي تكون من تدرجات الرمادي أو تكون بالأبيض والأسود وذلك لأنها تكون فعالة جداً عند وجود تكرارات كثيرة للقيم. تعتبر هذه الخوارزمية جزء من خوارزمية JPEG في ضغط الصور.بفرض لدينا الصورة التالية التي تحوي ألوان من تدرجات الرمادي:

12 12 12 12 12 12 12 12 12 35 76 112 67 87 87 87 5 5 5 5 5 5 1 .....

بنفس الطريقة نكتب عدد مرات التكرار قبل القيمة المكررة فتصبح الصورة على الشكل:

9 12 35 76 112 67 3 87 6 5 1

واجهتنا نفس المشكلة في ضغط النصوص وهو كيف يمكننا أن نميز بين قيمة معينة هل هي من الصورة أو قيمة لتكرار؟

الحل هو أن نقوم بإضافة Byte زائد لكل byte في الصورة مهمته هو التمييز بين معلومات الصورة وقيم التكرار. نقابل كل Bit من ال Byte الزائد مع كل Bit من ال Byte الخاص بالصورة, تكون قيمة الbit = 1 إذا كان مقابله في شكل ال Byteالزائد الأول Byte من الصورة السابقة على الشكل التالي:

9 12 35 76 112 67 3 87

1 0 0 0 0 0 1 0

يوجد طريقة ثانية لضغط الصور باستخدام خوارزمية RLE. هذه الطريقة فعالة إذا كانت الصورة ملونة بفرض لدي الصورة التالية:

B	B	B	A	A	A	A	A
C	C	C	C	B	B	B	B
B	B	B	D	D	E	D	D
B	B	B	B	B	D	D	D
B	B	B	B	E	E	B	B
B	B	B	B	E	E	A	B
B	B	B	B	B	B	A	A
B	B	B	B	B	A	A	A

فكرة هذه الطريقة هي تخزين الصورة على شكل ثنائيات (لون - تكرار) وتعمل بشكل "زيك زاك" أي يبدأ السطر الجديد من مكان انتهاء السطر الذي سبقه. في السطر الأول نجد انع يوجد ثلاث تكرارات للون B إذاً نضع الثنائية (b,3) وهكذا.

تتمثل هذه الطريقة بالشكل التالي:

(B,3) (A,5) (B,4) (C,4)

(B,3) (D,2) (E,1) (D,5)

(B, 9) (E,2) (B,3) (A,1)

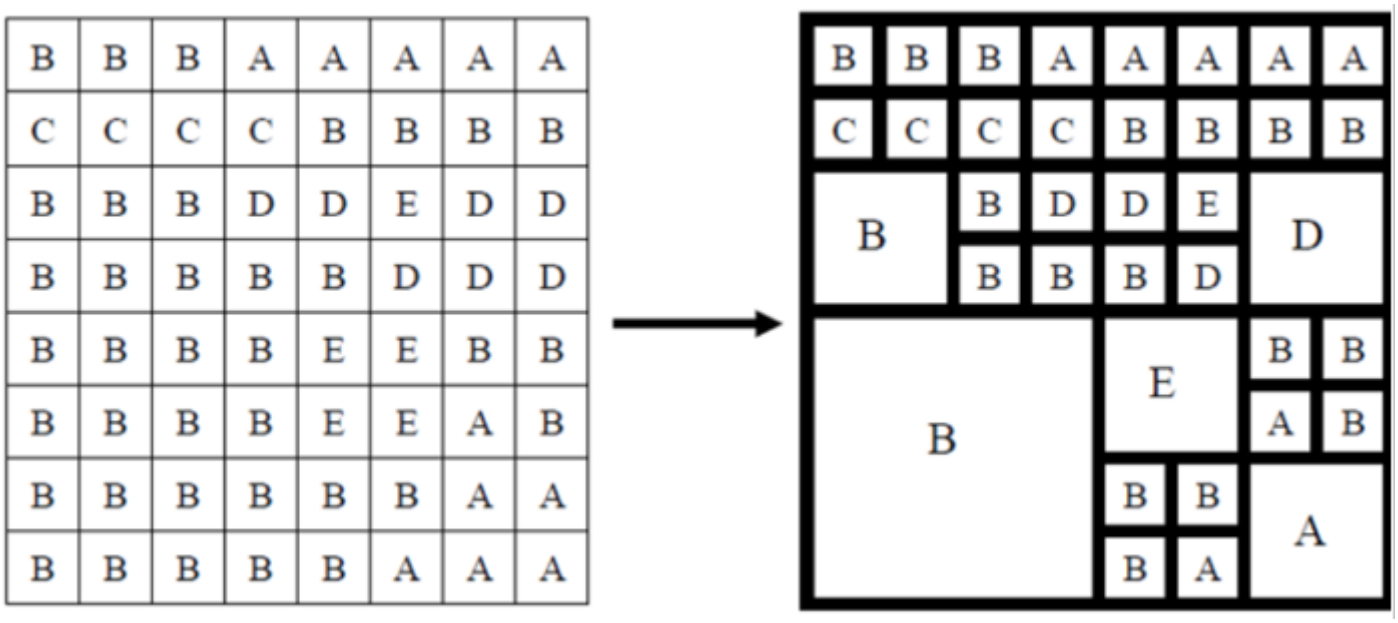
(E,2) (B,10) (A,5) (B,5)

إن حجم الصورة قبل الضغط كان عبارة عن (طول 8 * عرض 8) 64 مضروبة بقيمة تخزين القيمة اللونية الواحدة في الصورة. بينما حجم الصورة المضغوطة هو (16 قيمة لونية + 16 قيمة للتكرار) 32 مضروبة بقيمة تخزين الثنائية.

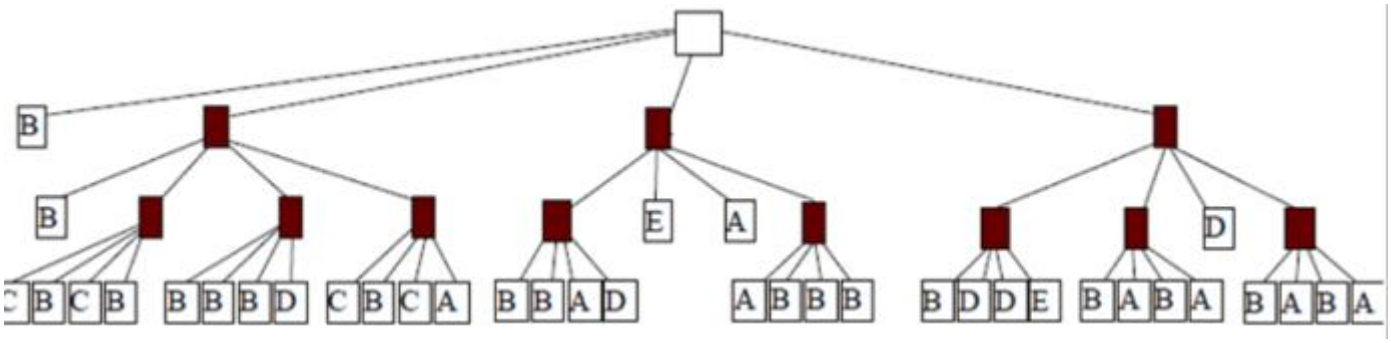
الطريقة الثالثة والأخيرة لضغط الصور باستخدام هذه الخوارزمية هي طريقة QuadTree

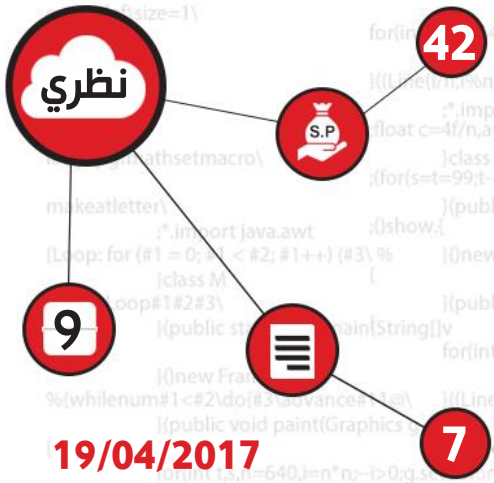
تعتمد هذه الطريقة على التجانس الطبيعي للجوار حيث تقوم بتقسيم الصورة إلى أربعة أقسام والقسم المتجانس(**أي كل القيم اللونية الموجودة في هذا القسم هي نفسها**) يتم تخزينه على أنه خانة واحدة والقسم غير المتجانس تقوم بتقسيمه إلى أربعة أقسام أخرى وهكذا حتى نصل إلى شجرة تكون فيها كافة الأقسام متجانسة أو نصل إلى حالة لا نستطيع التقسيم. ثم نقوم بتخزين هذه الشجرة لتعبر عن الصورة.

نطبق هذه الطريقة على الصورة السابقة :



ويكون شكل الشجرة كما هو موضح:





ضغط الصور والملفات 2

م.عمر سليمان

19/04/2017

في المحاضرة السابقة شرحنا بعض خوارزميات الضغط التي تعتمد على التكرار مثل RLE, لكن كما نعلم أن التكرار في النصوص الحقيقية قليل وكنه وارد كثيراً في الصور, سنشرح في هذه المحاضرة خوارزمية أفضل لضغط النصوص وهي خوارزمية الترميز الحسابي Arithmetic Coding التي تعتبر نوع من Lossless Compression أي يتم الضغط دون فقدان أي معلومة.

مبدأ هذه الخوارزمية هي استبدال تكرارات المحرف الواحد كلها برقم فاصلة عائمة وحيد, خرج هذه الخوارزمية هو عبارة عن هذا الرقم الذي تتراوح قيمه بين ال 0 وال 1, باستخدام خوارزمية معاكسة لفك الضغط نستطيع إرجاع هذا الرقم إلى تكرارات المحرف التي أنشئته.

للحصول على هذا الرقم السحري يجي علينا القيام بإحصائيات لمعرفة عدد مرات ورود محرف ما في النص, ثم حساب احتمال ورود هذا المحرف في النص كله, وذلك بتقسيم عدد مرات الورود على عدد محارق النص فنحصل على احتمال ورود المحرف في النص, بإدخال قيمة الإحتمال هذه إلى خوارزمية الضغط نحصل على الرقم السحري لهذا المحرف, ثم نطبق خوارزمية لفك الضغط على هذا الرقم فنحصل على النص الأصلي.

مثال:

بفرض لدينا النص التالي المؤلف من كلمتين ونريد ضغطه باستخدام خوارزمية الترميز الحسابي:

SWISS MISS

نقوم أولا ببناء جدول الإحتمالات للمحارف الواردة في النص, نقوم بحساب عدد تكرار كل محرف ثم حساب الإحتمالية, بعد تطبيق هذه الخطوة على النص السابق نحصل على الجدول التالي:

Char	Freq.	Prob.	Range
S	5	5/10=0.5	[0.5, 1.0)
W	1	1/10=0.1	[0.4, 0.5)
I	2	2/10=0.2	[0.2, 0.4)
M	1	1/10=0.1	[0.1, 0.2)
space	1	1/10=0.1	[0.0, 0.1)

العمود الأخير من الجدول السابق هو عبارة عن الفضاء الإحتمالي لاحتمالات ورود المحارف.

ملاحظة:

طريقة حساب القيم للجدول:

- **للعمود Prob.: نقوم حساب عدد مرات تكرار الحرف الذي نعمل عليه ونقسمع على عدد الأحرف جميعها**

الحرف S تكرر 5 مرات في النص أي ان نصف الجملة هي من هذا الحرف لذلك قيمته 0.5

((يجب أن يكون مجموع القيم لهذا العمود تساوي 1)).

- **للعمود Range: نضع في السطر الأول أن القيمة تبدأ من احتمال المحرف حتى القيمة 1. في السطر الذي يليه يصغر المجال ليصبح من قيمة المحرف حتى القيمة الصغرى للمجال الذي يسبقه وهكذا.**

الآن بعد الحصول على الجدول نبدأ بتطبيق الخوارزمية:

```
low = 0.0;
high = 1;
while (( c = getc( input )) != EDF )
{
      range = high – low;
      high = low + range * high_range(c);
      low = low + range * low_range(c);
}
Output(low);
```

تبدأ الخوارزمية بتعريف متحولين low, high وإعطائهم قيم ابتدائية, ثم نطبق الخوارزمية على كل محرف موجود في النص حيث نقوم بحساب قيم low, high الخاصة بكل محرف وتخزين هذه القيم في جدول, بالنسبة لمثالنا فبعد تطبيق الخوارزمية على المحارف نحصل على الجدول التالي:

C	Low_range(c)	High_range(c)	range	L	h
S	0.5	1	1	0.5	1
W	0.4	0.5	0.5	0.7	0.75
I	0.2	0.4	0.05	0.71	0.72
S	0.5	1	0.01	0.715	0.72
S	0.5	1	0.005	0.7175	0.72
	0	0.1	0.0025	0.7175	0.71775
M	0.1	0.2	0.00025	0.717525	0.71755
I	0.2	0.4	2.5E-05	0.71753	0.717535
S	0.5	1	5E-06	0.7175325	0.717535
S	0.5	1	2.5E-06	0.71753375	0.717535

بعد حساب قيم الجدول بتطبيق الخوارزمية (**قد يأتي سؤال عن تعبئة الخانات في الجدول بتطبيق الخوارزمية بشكل فعلي على نص**) نأخذ آخر قيمة وصلنا لها من عمود L (**بالأحمر**) وهذه القيمة تكون هي **الرقم السحري** الخاص بالنص الذي طبقنا عليه الخوارزمية.

لفك الترميز يوجد خوارزمية معاكسة للخوارزمية السابقة. حيث تكون دخلها المحارف المستخدمة في النص وجدول الفضاء الإحتمالي لهذا المحرف. باستخدام **الرقم السحري** الذي حصلنا عليه في عملية الترميز وتطبيق خوارزمية معاكسة أستطيع الحصول على النص الأساسي, خوارزمية فك الترميز:

```
number = input_code();
for ( ; ; )
{
      symbol = find_symbol_in_range( number );
      putc( symbole );
      range = high_range(symbol) – low_range(symbol);
      number = (number – low_range(symbole)) / range;
}
```

ضغط الصور:

شانون:

نرتب الرموز وفقاً للتردد بحيث تكون الرموز الأكثر شيوعاً من الجهة اليسرى.

تقسيم مجموعة الرموز إلى جزأين حيث مجموع ترددات المجموعة مقارب لترددات المجموعة الثانية.

تكرار العملية حتى الوصول إلى رمز واحد في المجموعة.

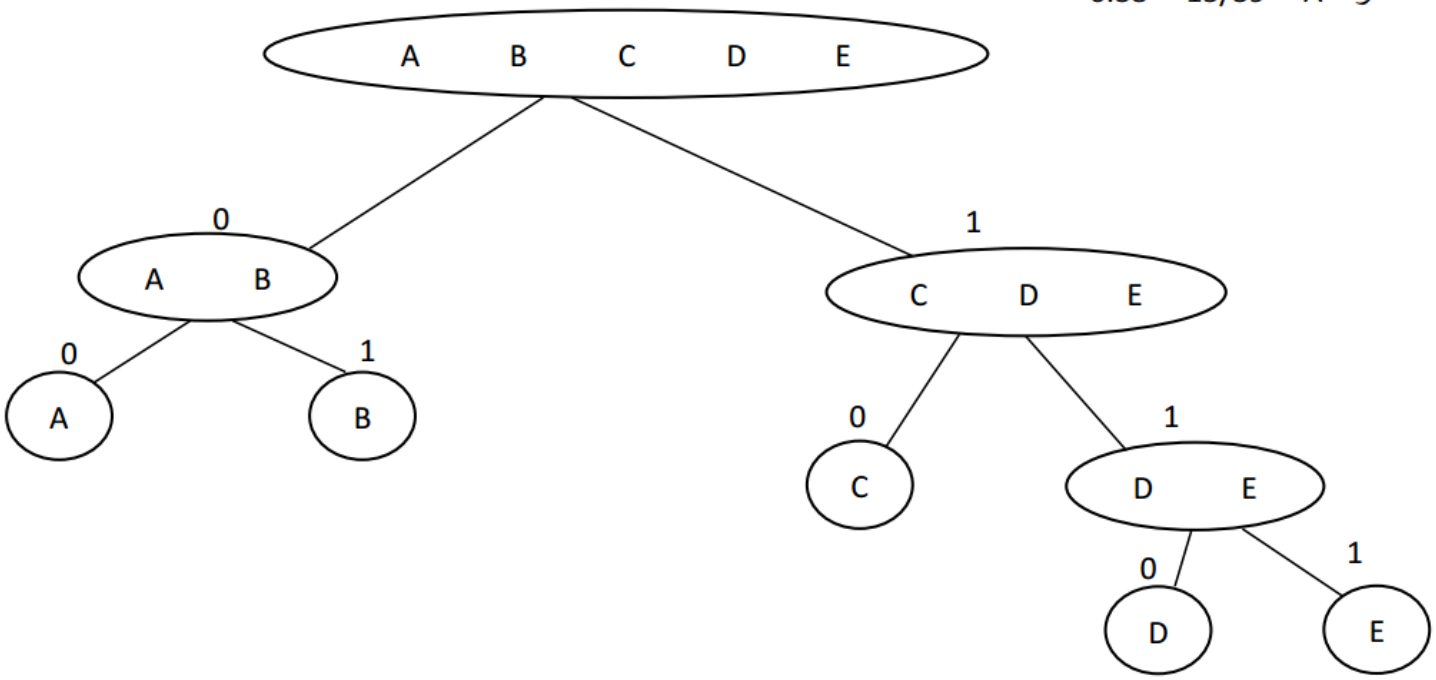
ترقيم الجزء الأيسر بالرقم 0 والجزء الأيمن بالرقم 1 أي أن الجهة اليسرى سوف تبدأ بصفر والجهة اليمنى بواحد.

الرمز	E	D	C	B	A
التكرار	5	6	6	7	15
التردد	0.128	0.15	0.15	0.179	0.38

النص غير موجود لكن نحن نعلم عدد تكرارات الأحرف فيه وإن عدد أحرفه يساوي 39. (مجموع التكرارات)
من خلال التقسيم أوجدنا التردد.

نحن بحاجة إلى ضرب مجموعة تكرار كل محرف بالرقم 8 للحصول على الحجم بال bit.

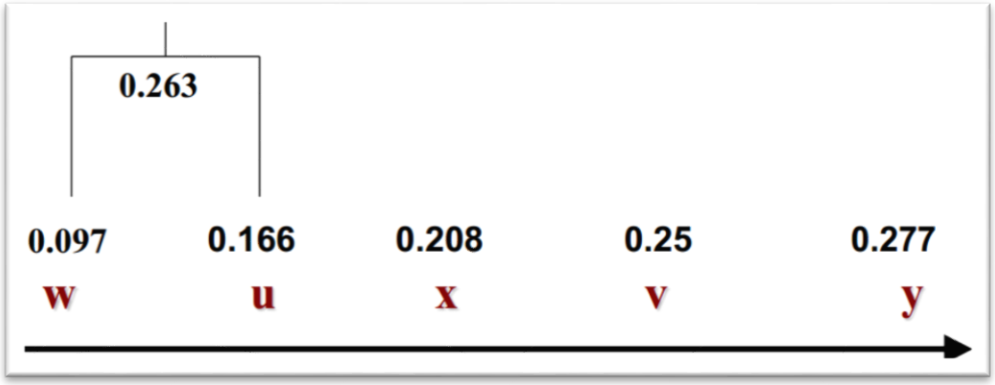
تردد A = 15/39 = 0.38



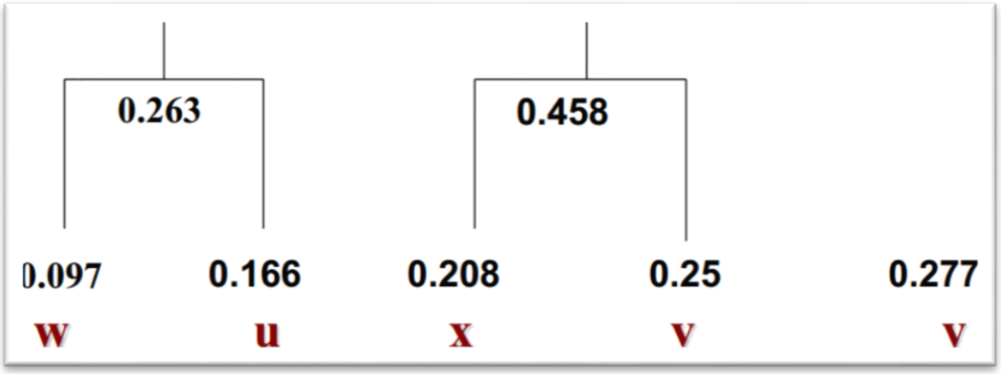
A: 00 - B: 01 - C:10 - D: 110 - E: 111

هوفمان:

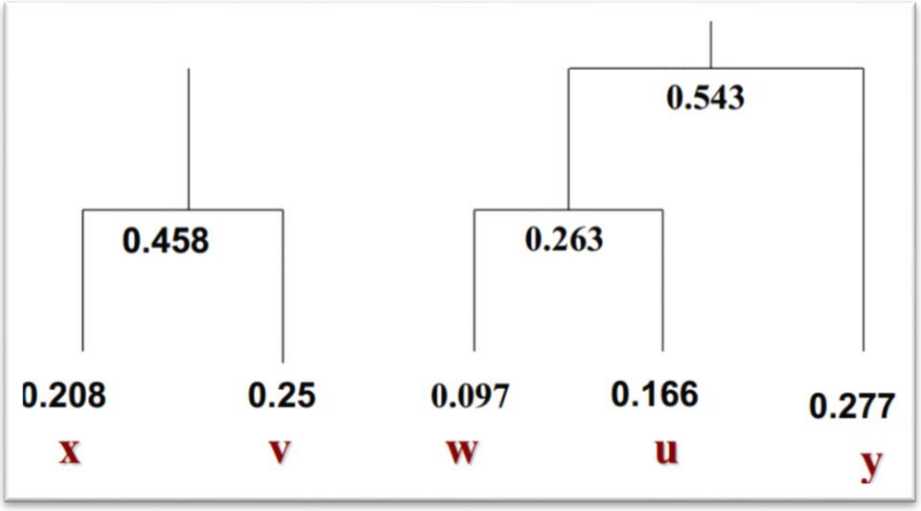
Symbol	Repeat	Probability	Data size
“w”	7	0.097	56
“u”	12	0.166	96
“x”	15	0.208	120
“v”	18	0.25	144
“y”	20	0.277	160
Sum	72	1	576



قمنا هنا بجمع أول قيمتين:

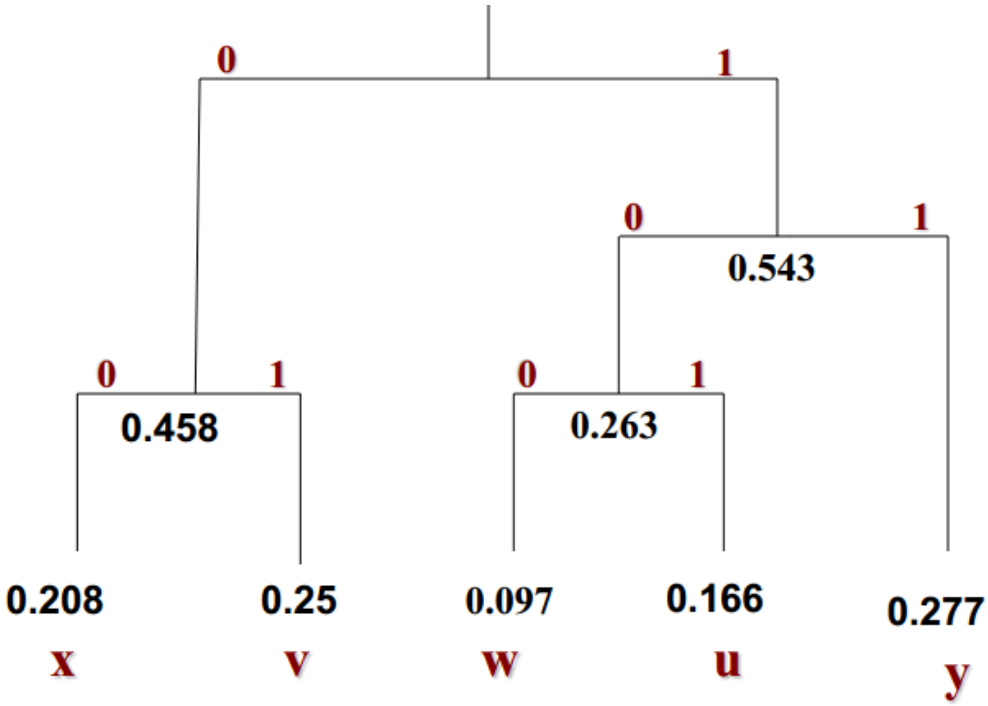


قمنا هنا بجمع القيم التي تليها:



قمنا هنا بجمع اخر قيمة مع مجمع سابقتها:

الناتج لدينا:



x= 0 0

v= 0 1

w= 10 0

u= 10 1

y= 1 1

لحساب الحجم الجديد : 15x2 + 18x2 + 7x3 + 12x3 + 20x2 = 163 bits

وبهذا قد قمنا بالضغط بنسبة : 1 - (163/576) = 71.7%

الصوت:

معلومات الصوت هي معلومات ذات بعد وحيد وهو الزمن † بينما الصورة هي ذات بعدين كلاهما هو موقع أي نحتاج لثنائية من موقع لتخزين الصورة (1, 1) أما الفيديو فهو ذو ثلاث أبعاد (صورة وزمن).

ملاحظة: الصورة نستطيع تجميدها لأنها مستقلة عن الزمن أما الصوت لا نستطيع.

الصوت: هو عبارة عن موجة اهتزازية تنتقل بواسطة الهواء نعتمد على الشدة والنبرة والتردد لتمييز الأصوات وإدراكها, يتم تخزين الصوت في مصفوفة أحادية البعد, يتم تحويل الصوت من إشارة تمثيل Analog إلى إشارة رقمية Digital بإستخدام ما يسمى الخطوة Step حيث يتم تقطيع الإشارة التماثلية حسب هذه الخطوة. كلما كان التقطيع قليل يكون الأداء افضل. في MP3 تقطيع قليل جداً, بينما إذا اردنا جودة ودقة عالية فيكون تقطيع كثير والخطوة صغيرة.

$$1Hz = \frac{1cycle}{second}$$

لرقمنة الصوت نحتاج إلى ثلاثة أشياء رئيسي:

1. معدل اخذ العينات: يعبر عن عدد العينات بالثانية.
2. دقة الصوت: يعبر عن عدد البتات المستخدمة لتمثيل كل عينة.
3. عدد القنوات: يكون عدد القنوات يساوي 1 في Mono, ويساوي 2 في Stereo.

لضغط الصوت نستخدم خوارزمية الترميز الحسابي: بسبب أن الصوت اقرب للنص, التكرار فيه نادر.

كيف تميز جهة الصوت باستخدام Stereo: الجهات تتحدد بهذا النمط بسبب ظهور الصوت على احدى القنوات قبل القناة الأخرى وعند تحسس الأذن لهذا الأمر فإن الدماغ يقوم بفهم الصوت على انه من جهة واحد.

للإتطلاع:

Quality	Sample Rate (KHz)	Bits per Sample	Mono/ Stereo	Data Rate (uncompressed) (kB/sec)	Frequency Band (KHz)
Telephone	8	8	Mono	8	0.200-3.4
AM Radio	11.025	8	Mono	11.0	0.1-5.5
FM Radio	22.05	16	Stereo	88.2	0.02-11
CD	44.1	16	Stereo	176.4	0.005-20
DAT	48	16	Stereo	192.0	0.005-20
DVD Audio	192 (max)	24 (max)	6 channels	1,200.0 (max)	0-96 (max)

ملاحظات:

- في افلام السينما يتم تسجيل الصوت على 6 قنوات (2 يسار – 2 منتصف – 2 يمين).
- الأذن البشرية تستطيع تمييز الأصوات في المجال الترددي 20 hz -> 22 hz.
  الحشرات تسمع الأصوات التي اعلى من مجال الأذن البشرية.
- إذا قمنا بتشغيل ملف صوتي بدقة MP3 على DVD فالنتيجة لا تتغير. لانه عندما قمنا بضغط الملف وتحويله ل mp3 فقدنا معلومات لا يمكن استرجاعها.