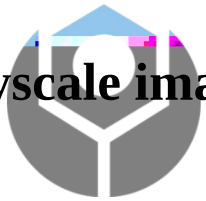
M9209213

Automatic grayscale image colorization



碩士學位論文指導教授推薦書

本校 資訊管理 系(所) 張佑瑋 君

所提之論文 自動灰階影像上色系統

係由本人指導撰述，同意提付審查。

指導教授 楊傳凱

九十四年七月十五日

碩士學位考試委員會審定書

本校 資訊管理 系（所） 張佑瑋 君

所提論文 自動灰階影像上色系統

，經本委員會審定通過，特此證明。

學位考試委員會

委　　員 李育杰

項天瑞

楊傳凱

指導教授：楊傳凱

系主任（所長）：楊維寧代

中華民國 九十四 年 七 月 十五 日

Welsh

Welsh

3×3 5×5

(tree-structure vector quantization)

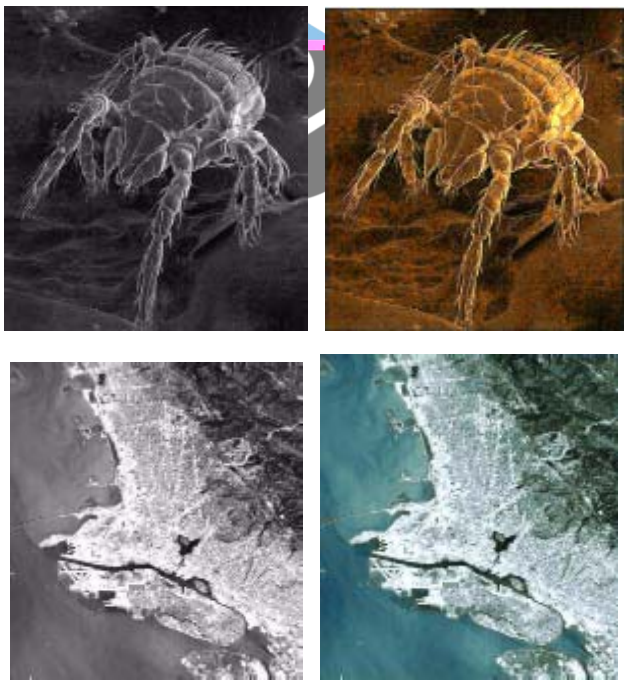
L_2 norm

1.1

x

(x-ray)

1-1



1-1

Welsh (transferring color to grayscale images) [1] Welsh

(wavelet transform)

[8] Welsh

1.2

Welsh[1] (Transferring Color to Greyscale Images)

(L_2 norm)

Welsh Reinhard

(Transferring Color Between Images) [2]

(target)

(source)

(texture synthesis)

Wei[3]

L_2 norm

Welsh

Welsh

(source) (target) Wei

Welsh Welsh 2002

Pentium800MHZ PC 15

4

Welsh

1.3

(pyramid)

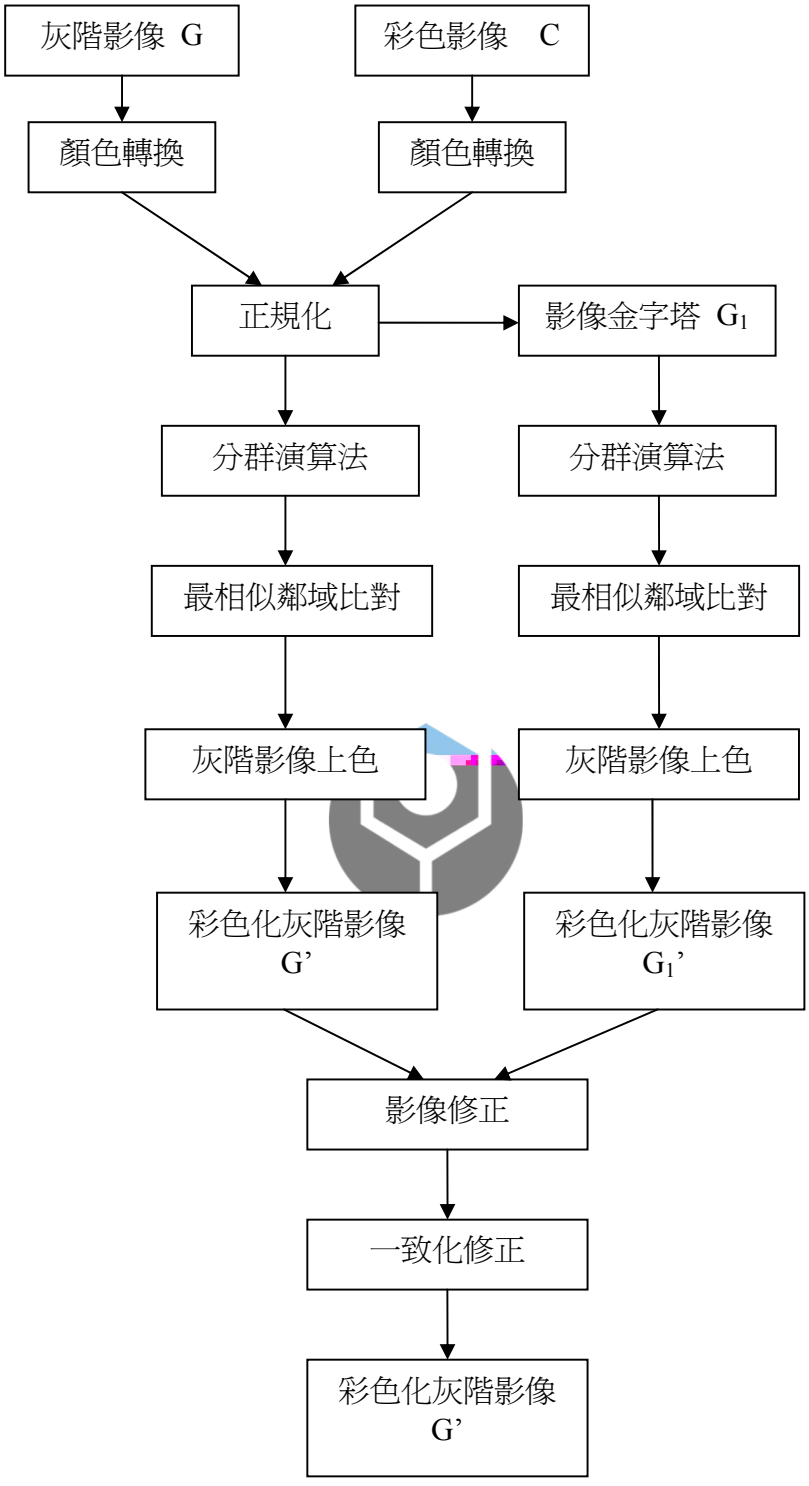
RGB *I α β* RGB

RGB 5×5 3

×3 (codebook)

9×9 3×3

:


灰階影像 G
彩色影像 C
顏色轉換
顏色轉換
正規化
影像金字塔 G_1
分群演算法
分群演算法
最相似鄰域比對
最相似鄰域比對
灰階影像上色
灰階影像上色
彩色化灰階影像 G'
彩色化灰階影像 G_1'
影像修正
一致化修正
彩色化灰階影像 G'

1-1

2.1

(

) RGB

2.1.1

Color transfer between images[2]

Reinhard et al.

2-1 2-1

(b) (a) (c)

2-1

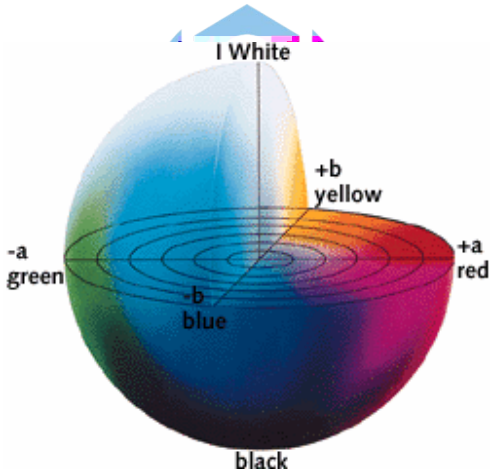
RGB RGB

Ruderman[2] /α β

/α β / α

( ) ( ) β ( ) ( )

2-2



2-2 /α β

2.1.2

Transferring Color to Grayscale Images[1]

Welsh et al.

2-3

(chromatic)

RGB Color Transfer Between Images [2] RGB $l\alpha\beta$

l

α β

L_2 norm (3.1.1)

(swatch)

2-3

2.1.3 Colorization using Optimization[5]

Levin (Colorization using Optimization)

(segmentation)

(track)

2-4(a) 2-4(b)

Levin (quadratic) (cost function)

(optimization)

(a) (b) (c)

(a) *(b)*

(c)

2-4

2.1.4

Fast Colorization of Gray Images [9]

Welsh

(antipole strategy)

k-mean

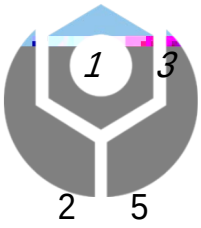
(TSVQ)

:

1. 5x5 25

2. 5x5 (mean) (standard deviation)

3. *1&2*



2~40 2 3

2

2.1.5

Color Transform for Digital Library Grey-value Images[8]

Welsh

Welsh

Welsh

c

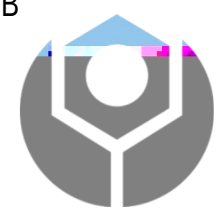
g

RGB $I\alpha\beta$

(luminance) α β

α β $I\alpha\beta$

RGB



L_2 norm (metric)

2.2

2.2.1

Fast Texture Synthesis using Tree-Structure Vector Quantization [4]

Wei

(Markov Random Field)

L

-

(tree-structured vector quantization)

L

$O(n^2)$ $O(\log n)$

(pyramid structure)

(tileability)

2.2.2

Synthesizing Natural Textures [6]

Michael Ashikhmin Wei[4]

Ashikhmin ,

L

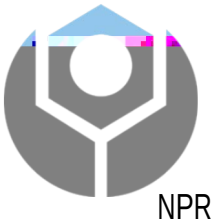
L

Wei[4]

Wei

2.2.3

Image Analogies [3]



(machine learning) NPR(non-photorealistic rendering)

(super-resolution) (texture transfer)

Hertzmann Wei[4] Ashikhmin[6]

A A’ B A A’ A

A’ B B’ A: A’=B: B’ 2-5

Wei Ashikhman A A’

B A A

A’ B B’

(A A’) B

B’ ,

A: A:: B:

B B

[10]

2-6 A: A’ B

B’

A A’ B B’

2-5

A

A’

B

B’

2-6

3.1

Welsh[1]

Welsh

Welsh 5×5

9×9 3×3

(pyramid)

5×5 3×3

9×9 3×3

( )

(gradient)

3.1.1

Welsh

3.1.1

3.1.1 L2 Gradient

L2 norm

N

L2 norm

$$E(N_c, N_g) = \sum_{p \in N} (C(p) - G(p))^2$$

E N_c (c)

N_g (g) L2 norm C G

p

L2norm L2norm

:

5	5	5
5	5	5
5	5	5

鄰域 N1

6	6	6
6	6	6
6	6	6

鄰域 N2

4	4	4
6	6	6
4	6	4

鄰域 N3

L_2norm E(N1,N2) E(N1,N3) 9

N2 N3 N1 N1 N2

L_2 norm

L_2norm (Gradient)[7]

:

$$G(N_q) = \sum_{p \in N} \left| I_p - I_q \right|$$

N q p q

q :

G(N1)=0 G(N2)=0, G(N3)=(6-6)+(6-4)+(6-6)+(6-6)=2

N1 N2

?

a b

0.5 abs(a-b)/(a+b)>0.5

a>3b b>3a

3.2

(target)  (source)

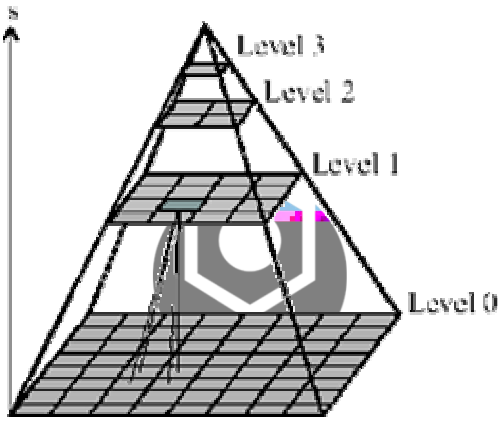
3.2.1

1/2

(Level 1)

A

B



3-1

3-1 Level 0

Level 1

1/4

3.2.2

( )

RGB /α β /α β /

(luminance) α β :

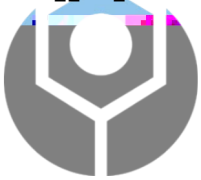
$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0708 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$L = \log L$$

$$M = \log M$$

$$S = \log S$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$



$$C(p) \leftarrow \frac{\sigma G}{\sigma C}(C(p) - mC) + mG$$

$C(p)$ mC σC

mG σG

?

3.2.3 (Tree Structure Vector Quantization)

(9×9 5×5 3×3)

(3×3

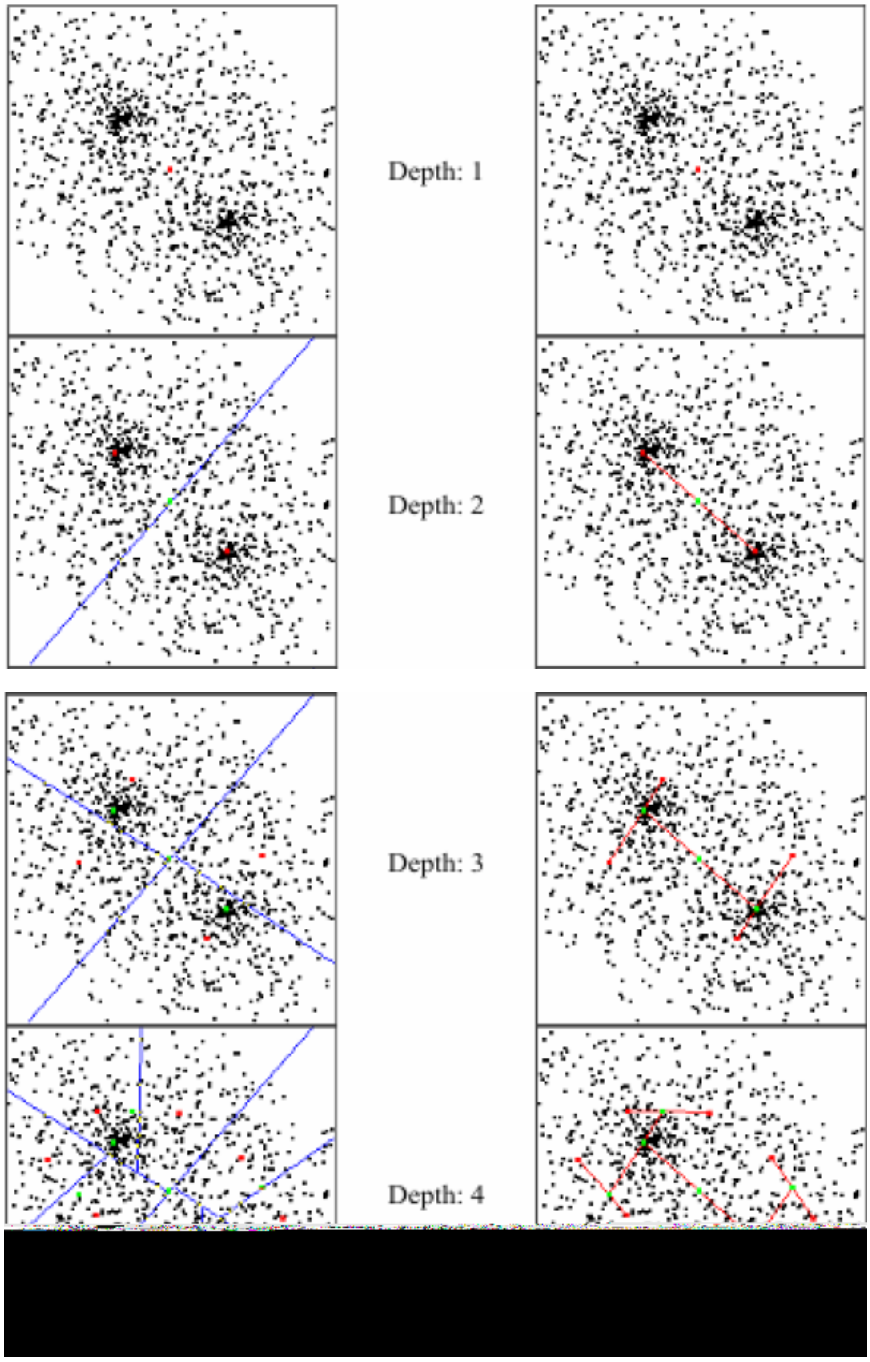
)

3×3 3×3

Lloyd Algorithm

(codebook)

3-2



3-2 TSVQ

$O(n^2)$ $O(\log n)$

[11]:

I_i	
G_i	
$G(L)$	L
$G(L,x,y)$	L (x, y)
T	

Function T ← BulidTree(G_i,L)

1. S ← null;
2. **loop** through all pixels (x_i,y_i) in the neighborhood is only texture of G_i
3. S ← S + BuildNeighborhood(G_i,L, x_i,y_i);
4. T ← InitializeTree(S);
5. T ← ComputeCentroids(T , Depth);
6. **return** T;

```
Function T ← ComputeCentroids(T , Depth)
    1. if (Depth    0) & (T repsrsents more than one vector of S)
    2.       (LeftChild(T ),RightChild (T )) ← PerturbedCentroids(T );
    3.       Dist ← Distortion(LeftChild(T ), RightChild(T ));
    4.       PastDist ← ValueMax;
    5.       while (PastDist    Dist) / PastDist > LloydThreshold
    6.             (LeftChild( T ),RightChild(T )) ←
                          ReCompute(LeftChild(T ), RightChild(T ));
    7.              PastDist ← Dist;
    8.              Dist ← Distortion(LeftChild(T ), RightChild(T ));
    9.  LeftChild(T ) ← ComputeCentroids(LeftChild(T ),
              Depth    1);
   10.  RightChild(T )← ComputeCentroids(RightChild(T ),
              Depth    1);
```

3.2.4

TSVQ

Level 1 Level 0 Level 1

TSVQ 5×5 3×3

Level 1 5×5

5×5 TSVQ

RGB

/α β RGB

/α β

α β α β RGB

(

) 3×3 3×3 TSVQ

α β α β

3

×3

Level 0 Level 1

Level 0

9×9 3×3 TSVQ

/α β RGB :

$$\begin{bmatrix} l \\ m \\ s \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix}$$

$$L = 10^{l}$$

$$M = 10^{m}$$

$$S = 10^{s}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

3.2.5

Level 0

Level 0 Level 1

3×3

3-3

0~4

1~4

:

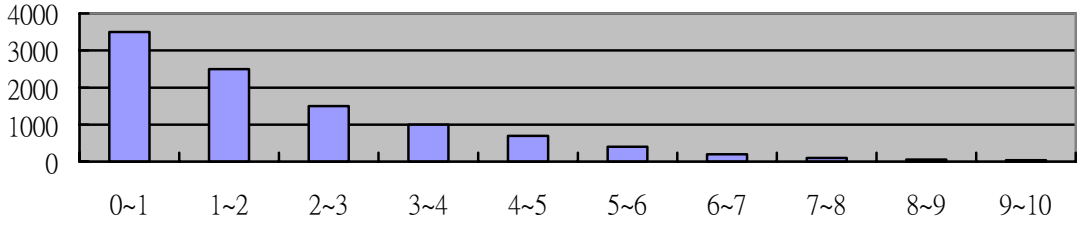
(3-4)

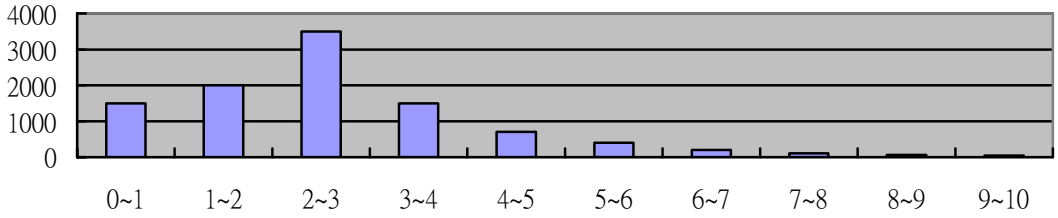
RGB 100×100 10%

10% 1000

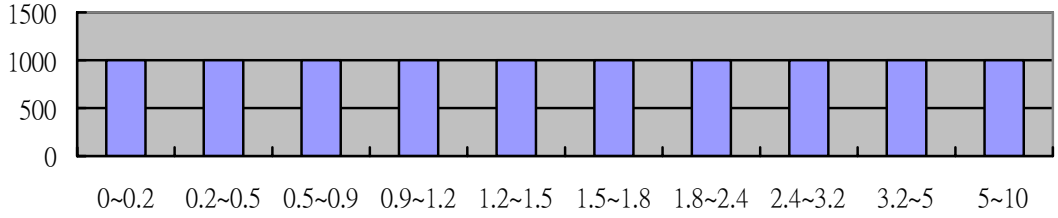
RGB / α β α β α β

/ α β RGB


4000
3000
2000
1000
0
0~1
1~2
2~3
3~4
4~5
5~6
6~7
7~8
8~9
9~10


4000
3000
2000
1000
0
0~1
1~2
2~3
3~4
4~5
5~6
6~7
7~8
8~9
9~10

3-3


1500
1000
500
0
0~0.2
0.2~0.5
0.5~0.9
0.9~1.2
1.2~1.5
1.5~1.8
1.8~2.4
2.4~3.2
3.2~5
5~10

3-4

3-5

1%

a

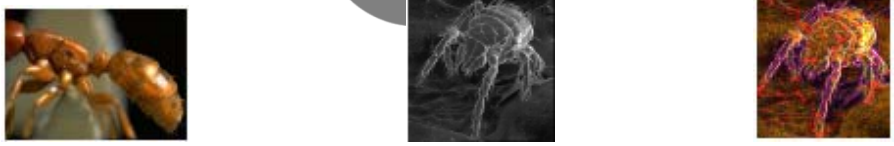
b level 1

b a c level 0

c a b

d

a



level 1 level 1 b

c d

3-5

4.1

:

	Pentium IV 2.53 GHz
	512MB
	Windows XP
	Visual C++.NET

4-1

4.2

(Level 0)

; (Level 0)

Welsh[1]

A_1 A_2

Welsh A_3 A_4

B_1 B_2



Welsh B_3 B_4 Blasi B_5


C_1 C_2



Welsh C_3 C_4 Blasi C_5

4-1

Welsh

Welsh A B C

(4-1) 5×5

5×5

Welsh A_3

B

B_4 Welsh Blasi B_5

C

Blasi C_5 Welsh

5x5 4-2 1

2

3

1 2

3

4-2 Welsh

D_3 (4-3)

? D_2

L_2 norm

D_4

D_1 D_2

Welsh D_3 D_4

4-3 1

E_1 E_2

Welsh E_3 E_4

4-4 2

4-4 E_2

E_3 L_2 norm

G_1 G_2

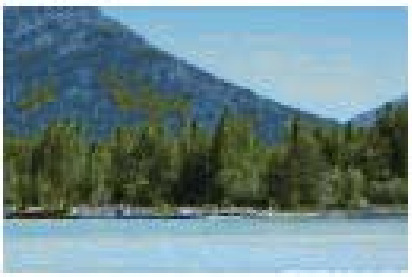
Welsh G_3 G_4

4-6 4

4-6 G_1 G_3

Welsh

H_1 H_2



Welsh H_3 H_4

I_1 I_2

Welsh I_3 I_4

J_1 J_2

Welsh J_3 J_4

4-7 5

4-7 H_1 I_1 J_1 Welsh

I Welsh

Blasi 4-8 K

Blasi

L L_4

L_5

M_4

M_5

A_4

3x3

a b

a b

4-9

10 40

4-2

( )

	Level 1	Level 0				
A_4 227×146	5.33s	40.53s	0.13s	9.22s	55.20s	1385s
B_4 149×159	2.98s	21.55s	0.08s	4.22s	28.83	1560s
			0.09s		18.75s	611s
			0.05s	1.89s	14.56s	191s
			0.06s	2.11s	19.48s	191s
			0.05s	1.84s	14.06s	190s
			0.00	s	15.75s	822

Blasi

Blasi

Welsh

4-2 4-3 Blasi

(source)

[1] T. Welsh, M. Ashikhmin, and K. Mueller, “TransferringColor to Greyscale Images”, Proceedings of SIGGRAPH 2002, July 2002.

[2] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, “Color Transfer between Images”, IEEE Computer Graphics and Applications, pp. 34-40, September/October 2001.

[3] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless and D. H. Salesin, “Image analogies”, Proceedings of SIGGRAPH 2001, July 2001.

[4] Li-Yi Wei and Marc Levoy, “Fast Texture Synthesis Using Tree-Structured Vector Quantization”, Proceedings of SIGGRAPH 2000, pages 479–488, July 2000.

[5] A. Levin, D. Lischinski and Y. Weiss, “Colorization using Optimization”,Proceedings of SIGGRAPH 2004, July 2004.

[6] Michael Ashikhmin, “Synthesizing Natural Textures”, 2001 ACM Symposium on Interactive 3D Graphics, pages 217–226, March 2001.

[7] Perez, P., Gangnet, M., Blake, A. “Poisson image editing”. SIGGRAPH (2003) 313–318

[8] , , , “Color Transform for Digital Library Grey-value Images”,

[9] G. Di Blasi, D. Reforgiato, “Fast Colorization of Gray

Images . In Proceedings of Eurographics Italian Chapter 2003.
[10] Jari Huttunen, "Image Analogies", Helsinki University of Technology,2004
[11] Paul Billault. "Texture Synthesis Algorithms". Ecole Polytechnique, Promotion X-9