Subject: Re: Time for little quiz!
Posted by mirek on Wed, 05 Apr 2006 08:53:10 GMT
View Forum Message <> Reply to Message

That would be much more code to write. In fact, there is a couple of methods in class, and class intance here is a sort of replacement of Pascal's embedded function/procedures.

BTW, as this is quite nice implementation of median cut palette algorithm, and I was trying quite hard to google some code for this, maybe placing it here will make further implementors search easier:

Median cut palette algorithm in C++:

```
struct sPalMaker {
 int        histogram[RASTER_MAP_R][RASTER_MAP_G][RASTER_MAP_B];
 int        colorcount;
 struct Dim {
  int l, h;

  operator int() { return h - l; }
 };
 enum { G = 0, R = 1, B = 2 };
 void Copy(Dim *d, Dim *s)   { d[R] = s[R]; d[G] = s[G]; d[B] = s[B]; }
 struct Box {
  int     volume;
  int     colorcount;
  int     population;
  Dim     dim[3];
  int     avg_r, avg_g, avg_b;

  Dim&    operator[](int i)    { return dim[i]; }
 };

 void Update(Box& box, int ii);

 sPalMaker(const RGBA *s, int count, RGBA *palette, int ncolors);
};

void sPalMaker::Update(Box& x, int ii)
{
 x.colorcount = 0;
 x.population = 0;
 int a[3][RASTER_MAP_MAX];
 ZeroArray(a[R]);
 ZeroArray(a[G]);
 ZeroArray(a[B]);
 x.avg_r = x.avg_g = x.avg_b = 0;
```

```
 for(int r = x[R].l; r < x[R].h; r++)
  for(int g = x[G].l; g < x[G].h; g++)
   for(int b = x[B].l; b < x[B].h; b++) {
    int q = histogram[r][g][b];
    a[R][r] += q;
    a[G][g] += q;
    a[B][b] += q;
    x.avg_r += q * r;
    x.avg_g += q * g;
    x.avg_b += q * b;
    x.colorcount += (-q >> 31) & 1;
    x.population += q;
   }
 for(int i = 0; i < 3; i++) {
  Dim& d = x[i];
  while(d.l < d.h && a[i][d.l] == 0)
   d.l++;
  while(d.h > d.l && a[i][d.h - 1] == 0)
   d.h--;
 }
 x.volume = x[R] * x[G] * x[B];
}

static byte sRc(int avg, int pop, int div)
{
 return Saturate255(255 * (avg + (pop >> 1)) / pop / (div - 1));
}

sPalMaker::sPalMaker(const RGBA *s, int count, RGBA *palette, int ncolors)
{
 ASSERT(ncolors <= 256);
 ZeroArray(histogram);
 for(const RGBA *e = s + count; s < e; s++)
  histogram[s->r >> RASTER_SHIFT_R][s->g >> RASTER_SHIFT_G][s->b >>
RASTER_SHIFT_B]++;
 Buffer<Box> box(256);
 box[0][R].l = 0;
 box[0][R].h = RASTER_MAP_R;
 box[0][G].l = 0;
 box[0][G].h = RASTER_MAP_G;
 box[0][B].l = 0;
 box[0][B].h = RASTER_MAP_B;
 Update(box[0], 0);
 if(box[0].population == 0)
  return;
 colorcount = box[0].colorcount;
 count = 1;
 int method = 1;
```

```
 while(count < ncolors) {
  int ii = -1;
  int maxv = 0;
  if(2 * count > ncolors)
   method = 1;
  for(int i = 0; i < count; i++) {
   int v = method ? box[i].volume : box[i].colorcount;
   if(box[i].colorcount > 1 && v > maxv) {
    ii = i;
    maxv = v;
   }
  }
  if(ii < 0)
   break;
  Box& b = box[ii];
  int ci = b[R] > b[G] ? b[B] > b[R] ? B : R : b[B] > b[G] ? B : G;
  if(b[ci] == 1) {
   if(method == 1)
    break;
   method = 1;
  }
  else {
   int m = (b[ci].l + b[ci].h) >> 1;
   Box& b1 = box[count];
   b1 = b;
   b[ci].h = m;
   b1[ci].l = m;
   Update(b, ii);
   Update(b1, count++);
  }
 }
 for(int i = 0; i < count; i++) {
  RGBA& c = palette[i];
  Box& x = box[i];
  c.r = sRc(x.avg_r, x.population, RASTER_MAP_R);
  c.g = sRc(x.avg_g, x.population, RASTER_MAP_G);
  c.b = sRc(x.avg_b, x.population, RASTER_MAP_B);
  c.a = 255;
 }
}

void CreatePalette(const RGBA *s, int count, RGBA *palette, int ncolors)
{
 ASSERT(ncolors <= 256);
 delete new sPalMaker(s, count, palette, ncolors);
}
```