
Subject: stl question

Posted by [GaroRobe](#) on Fri, 10 Jun 2011 03:37:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

So.

I've got a method in OpenCV I need to use. It has parameters with type `vector<vector<Point3d>>`. It is a data about primitive regular mesh (in model coords) for each frame. Filling it manually feels wrong, so I do it like:

```
vector<Point2f> checkboardMesh;
```

```
for ( int i = 0; i < board_h; i++ )
  for ( int j = 0; j < board_w; j++ )
    checkboardMesh.push_back ( * new Point2f ( j, i ) );
```

And here I get stuck: will I not get memory leak here? AFAIK `std::vector` copies element on insert. Should I do it like

```
for ( int i = 0; i < board_h; i++ )
for ( int j = 0; j < board_w; j++ )
{
  tmpP = new Point2f ( j, i );
  checkboardMesh.push_back ( *tmpP );
  delete tmpP;
}
```

maybe? If I do, then I'd rather look for a better idea.

Or maybe there is some way of using Upp containers here?

P.S.: The method I mentioned is `cv::calibrateCamera` btw:

```
double cv::calibrateCamera( const vector<vector<Point3f> >& objectPoints,
                           const vector<vector<Point2f> >& imagePoints,
                           Size imageSize, Mat& cameraMatrix, Mat& distCoeffs,
                           vector<Mat>& rvecs, vector<Mat>& tvecs, int flags )
```

Subject: Re: stl question

Posted by [dolik.rce](#) on Fri, 10 Jun 2011 05:55:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi GaroRobe,

You don't have to use `new` at all. Just creating a temporary object is fine, since `std::vector` will make a copy:

```
vector<Point2f> checkboardMesh;
```

```
for ( int i = 0; i < board_h; i++ )
  for ( int j = 0; j < board_w; j++ )
    checkboardMesh.push_back ( Point2f ( j, i ) );
```

Using U++ containers should be possible, but it would be an ugly hack. Something along the lines of: using std::vector;
using cv::Point2f;

```
Vector<Point2f> v; //this requires to do NAMESPACE_UPP; NTL_MOVEABLE(cv::Point2f);  
END_UPP_NAMESPACE; in global scope
```

```
vector<Point2f> checkboardMesh;
```

```
for ( int i = 0; i < board_h; i++ )  
  for ( int j = 0; j < board_w; j++ )  
    v.Add( Point2f ( j, i ) );
```

```
checkboardMesh.resize(v.GetCount());  
memmove(&checkboardMesh[0],v.Begin(),v.GetCount()*sizeof(Point2f));
```

In other words: not elegant and definitely not recommended. Please never do something like that

Honza

Subject: Re: stl question
Posted by [GaroRobe](#) on Fri, 10 Jun 2011 06:21:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wouldn't have guessed. Thanks a lot
