reprojectImageTo3D()

```
void cv::reprojectImageTo3D ( InputArray    disparity,
                              OutputArray   _3dImage,
                              InputArray    Q,
                              bool          handleMissingValues = false ,
                              int           ddepth = -1
                            )
```

Python:

```
cv.reprojectImageTo3D( disparity, Q[, _3dImage[, handleMissingValues[, ddepth]]] ) -> _3dImage
```

```
#include <opencv2/calib3d.hpp>
```

Reprojects a disparity image to 3D space.

Parameters

disparity Input single-channel 8-bit unsigned, 16-bit signed, 32-bit signed or 32-bit floating-point disparity image. The values of 8-bit / 16-bit signed formats are assumed to have no fractional bits. If the disparity is 16-bit signed format, as computed by **StereoBM** or **StereoSGBM** and maybe other algorithms, it should be divided by 16 (and scaled to float) before being used here.

_3dImage Output 3-channel floating-point image of the same size as disparity. Each element of _3dImage(x,y) contains 3D coordinates of the point (x,y) computed from the disparity map. If one uses Q obtained by **stereoRectify**, then the returned points are represented in the first camera's rectified coordinate system.

Q 4×4 perspective transformation matrix that can be obtained with **stereoRectify**.

handleMissingValues Indicates, whether the function should handle missing values (i.e. points where the disparity was not computed). If handleMissingValues=true, then pixels with the minimal disparity that corresponds to the outliers (see **StereoMatcher::compute**) are transformed to 3D points with a very large Z value (currently set to 10000).

ddepth The optional output array depth. If it is -1, the output image will have CV_32F depth. ddepth can also be set to CV_16S, CV_32S or CV_32F.

The function transforms a single-channel disparity map to a 3-channel image representing a 3D surface. That is, for each pixel (x,y) and the corresponding disparity d=disparity(x,y) , it computes:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = Q \begin{bmatrix} x \\ y \\ \texttt{disparity}(x,y) \\ 1 \end{bmatrix}.$$

See also

To reproject a sparse set of points {(x,y,d),...} to 3D space, use **perspectiveTransform**.