Warping Functionality

The Warping Library supports the following functionality:

- Warp matrix computation
- Warp matrix adjustment
- Reference Point Shift Vector encoding and retrieval
- Warp image bounds computation

Warp Matrix Computation

Input:

- Shift Vectors
- Configuration information (please refer to Configuration of warping parameters)

Output:

- Warp Matrix

Algorithm:

- Decode and cache the reference point shift vectors.
- Computation of warp matrix from reference points and corresponding shift vectors. The reference points are distributed over the whole bitmap area according to the specifications in the parameter list.
- Rotation of warp matrix. The rotation of the warped reference points take place aroung the center point of the warped bitmap. (please refer to Rotation and Scale)
- Scaling of warp matrix to illuminated area, if outside of illuminated area. (please refer to Rotation and Scale)
- Normalize of warp matrix. Candera uses normalized coordinates in range [0..1] for rendering, therefore the warp matrix data has to be normalized in this range.

Interface:

CalculateWarpMatrix(const stc_warp_configuration_parameter_t* const pstcWarpConfigParams, const stc_warp_parameter_set_t* const pstcWarpRefPoints, stc_warp_matrix_t* pstcWarpMatrix)

Warp Matrix Adjustment

Input:

- Configuration information.
- Adjustment configuration (please refer to Adjustments).

Output:

- Warp Matrix

Algorithm:

- Adjustment of reference point shift vectors according to the specified adjustment configuration structure using the according mathematical formulas.
- Computation of warp matrix from reference points and corresponding shift vectors.
- Rotation of warp matrix.
- Scaling of warp matrix to illuminated area, if outside of illuminated area.
- Normalize of warp matrix.

Interface:

AdjustWarpMatrix(const stc_warp_configuration_parameter_t* const pstcWarpConfigParams, const stc_warp_adjust_parameter_t* const pstcWarpAdjustParams, en_warp_bool_t* pbRefPointsChanged, stc_warp_matrix_t* pstcWarpMatrix)

Reference Points Encoding and Retrieval

Input:

- Internally cached configuration
- Internally cached shift vectors

Output:

- Encoded bitmap image shift vectors

Algorithm:

- Encode shift vectors.
- Write shift vectors to application supplied memory location.

The current active reference point shift vector set is retrieved from the warping library. Shift vectors are encoded according to the provided configuration.

Interface:

GetRefPoints(stc_warp_parameter_set_t* pstcWarpRefPoints)

Warp Image Bounds Computation

Input:

- Configuration information.

Output:

- Bitmap image area bounds in normalized coordinates

Algorithm:

- Calculate area bounds based on the provided configuration information:
  - x = (bitmapAreaShiftX - bitmapAreaX/2) / displaySizeX
  - y = (bitmapAreaShiftY - bitmapAreaY/2) / displaySizeY
  - width = bitmapAreaX / displaySizeX
  - height = bitmapAreaY / displaySizeY

The image area bounds are normalized as required by Candera, as it uses normalized coordinates in range [0..1] for rendering.

Interface:

ComputeWarpImageBoundsFromConfig(stc_warp_configuration_parameter_t* config, WRP_SFLOAT* x, WRP_SFLOAT* y, WRP_SFLOAT* width, WRP_SFLOAT* height)

Revision #4
Created 2023-02-21 02:56:26 UTC by Kanai Tomoaki
Updated 2023-06-12 07:54:57 UTC by Tsuyoshi.Kato