Warping Library API

The public interface of the Warping Library is presented below:

Method	Parameters
en_warp_result_t **CalculateWarpMatrix()** • calculates the WarpMatrix from configuration parameters and reference points passed.	• stc_warp_configuration_pamameter_t* [IN] **pstcWarpConfigParams**: pointer to structure holding the configuration parameters and limits that need to be checked. • stc_warp_parameter_set_t* [IN] **pstcWarpRefPoints**: pointer to structure holding the packed reference points. These values are internally copied, so that the caller may change the values in the structure without any side effect to the reference points held by the library. • stc_warp_matrix_t* [OUT] **pstcWarpMatrix**: pointer to data structure in which the WarpMatrix is written to. Memory must be allocated and managed by the caller. > The values written into this data structure are valid only if success code is returned, otherwise data might be inconsistent or incomplete.
en_warp_result_t **AdjustWarpMatrix()** • adjust WarpMatrix from configuration parameters and adjust passed parameters based on internally stored reference points.	• stc_warp_configuration_pamameter_t* [IN] **pstcWarpConfigParams**: pointer to structure holding the configuration parameters and limits that need to be checked. • stc_warp_adjust_parameter_t* [IN] **pstcWarpAdjustParams**: pointer to structure holding the adjustment parameters. • stc_warp_bool_t* [OUT] **pbRefPointsChanged**: pointer to Boolean where the function returns a flag indicating if the internally stored reference points have been changed during the adjustments. Memory must be allocated and managed by the caller. • stc_warp_matrix_t* [OUT] **pstcWarpMatrix**: pointer to data structure in which the WarpMatrix is written to. Memory must be allocated and managed by the caller. > The values written into this data structure are valid only if success code is returned, otherwise data might be inconsistent or incomplete.

en_warp_result_t **GetRefPoints()** • copy the values of internal current and valid reference point to the data structure. **Note:** If any range check fails, appropriate error code is returned.	• stc_warp_parameter_set_t* [OUT] **pstWarpRefPoints**: pointer to data structure where the packed reference points will be copied to. Memory must be allocated and managed by the caller.
en_warp_result_t **ComputeWarpImageBoundsFromConfig()** • computes the bounds of the warping image within the full render target contents in normalized coordindates [0..1] from the warping configuration. **Note:** The bounds can be used as input for SetWarpImageBounds() function of Candera::Display.	• stc_warp_configuration_parameter_t* [IN] **config**: configuration used for warping. • WRP_SFLOAT* [OUT] **x**: coordinate of upper-most left bitmap area point. • WRP_SFLOAT* [OUT] **y**: coordinate of upper-most left bitmap area point • WRP_SFLOAT* [OUT] **width**: width of the bitmap area • WRP_SFLOAT* [OUT] **height**: height of the bitmap area

Revision #5
Created 2023-02-16 07:01:31 UTC by Tsuyoshi.Kato
Updated 2026-08-24 07:38:46 UTC by Kanai Tomoaki