

FAST OMNI-IMAGE UNWARPING ON THE JETSON TK1

Gustavo Silva, Jaime Reátegui, Paul Rodríguez

Pontificia Universidad Católica del Perú, Peru

[gustavo.silva, jaime.reategui, prodrig]@pucp.edu.pe


 PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

ABSTRACT

Omnidirectional sensors are very useful in different applications that require a wide field of view, such as robot vision, 3D reconstruction, video surveillance, etc. Computational time is an important factor for these applications, therefore there is the need for computationally efficient omni-image unwarping. In this work, an efficient and highly parallelizable algorithm that benefits from CUDA platform (Nvidia Jetson TK1) is proposed. The proposed method referred as pano-mapping offset table (PMOT) allows computation of the pano-image (from the omni-image) using the offset table. The PMOT is created once for an omni-camera calibration, pano-image resolution and interpolation method. Experiments on the Jetson TK1 show that the proposed method can achieve 1697 ~ 34 FPS for unwarping omni-images (512x512 ~ 4096x4096) into pano-images (128x512 ~ 1024x4096).

1. INTRODUCTION

CATADIOPTRIC omnidirectional vision sensors (omni-cameras) are becoming useful in applications that need a wide field of view (FOV), such as robot vision, 3D reconstruction, video surveillance, aerial vehicles, etc. The omni-camera provides an image that is called omni-image. This kind of image gives a field of view of 360 degrees, but it presents distortion due to the mirror of the catadioptric sensors [6]. Due to the significant distortions on the omni-image, classical image processing cannot be directly applied (digital image processing uses a uniform sampling). Omni-image unwarping is a method to solve this problem. It consists on transforming the omni-image into a pano-image. In this work, we propose an efficient and highly parallelizable alternative for unwarping omni-images referred as pano-mapping offset table (PMOT). Our method includes the following steps: first, PMOT is created, where each entry is an offset that permits relationship the omni-images and pano-image depending on the interpolation applied (nearest neighbors or bilinear interpolation). Finally using the PMOT, we unwarped the omni-image.

2. OBJECTIVE

The objective of this work is to provide a fast implementation (real-time implementation) for unwarping omnidirectional images on the Jetson TK1 platform.

3. STATE OF THE ART

In the past few years, several specific methods for omni-image unwarping have been proposed: [1] proposed a robust image registration using log-polar transform. [2] proposed a panoramic image transform using discrete geometry techniques (DGT). [3] proposed a pano-mapping table method for omni-image unwarping. This method outperforms the others in quality, complexity and processing time. [4] proposed using only one-eighth of the pano-mapping table for omni-image unwarping (for nearest neighbors interpolation) using geometry symmetry, reducing computational time due to fewer memory accesses. [5] proposed a pano-mapping pointers array method for omni-image unwarping (for nearest neighbors and bilinear interpolation). This array embeds all four steps [5] required for omni-image unwarping and outperforms the others methods in processing time.

3.1 Standard techniques to unwarped omni-images

The projection model is based on a unit sphere on which 3D points are projected [7]. The world point O_r is projected on to the unit sphere centered on C_m . The O_s is moved to a new center C_p and mapped onto a normalized plane. The final step involves computing the generalized camera projection [7]. These steps are shown in the next figure.

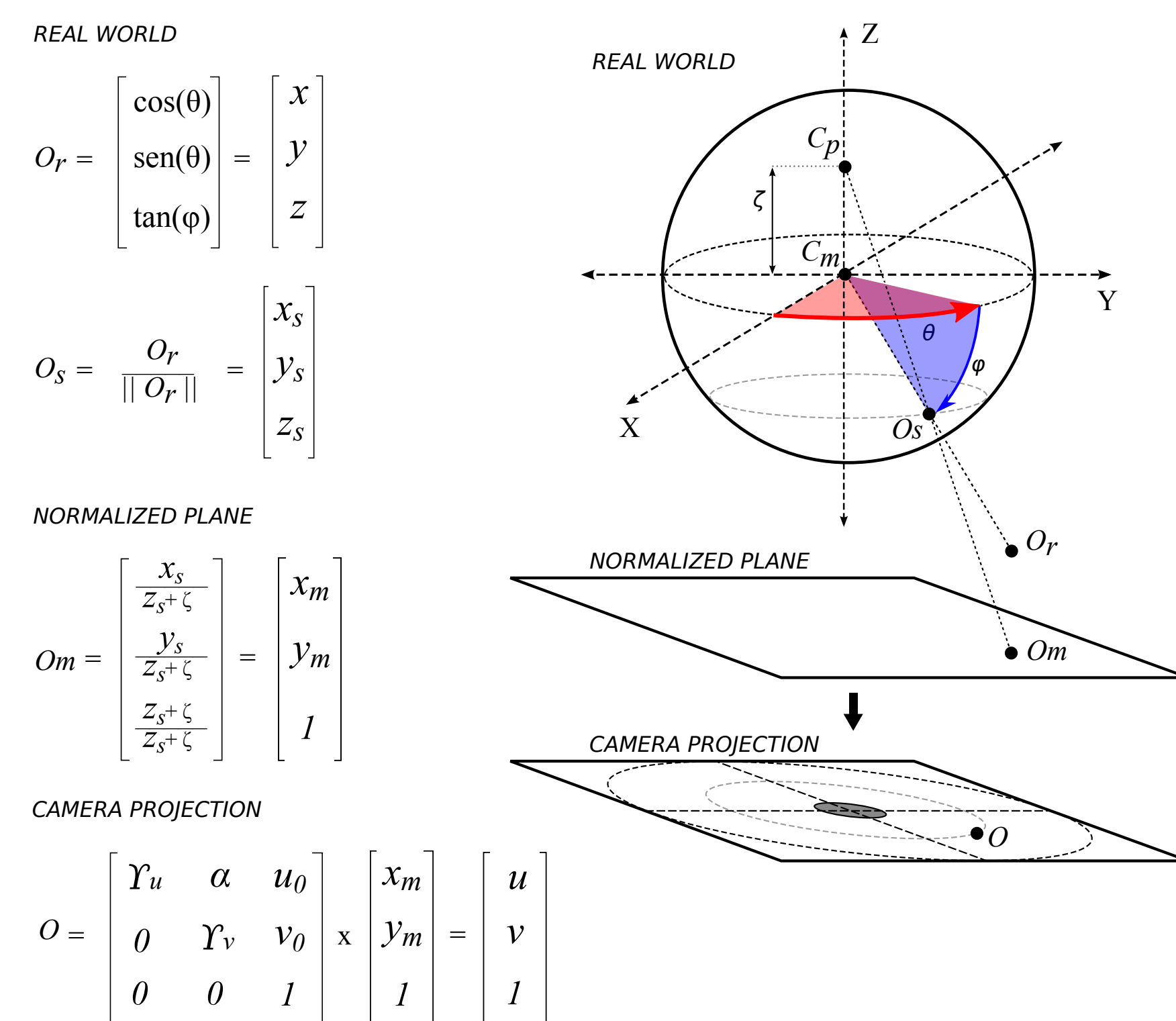


Figure 1: Standard projection model based on a unit sphere. O_r represent a real world point and O an omni-image point.

The parameter ζ is the distance between the center of the sphere and the focal point of omni-mirror. The parameters γ_u and γ_v are the horizontal and vertical focal lengths respectively, u_0 and v_0 are the coordinates of the principal point of the camera plane and α is the skew of both axis.

Pano-mapping table is built using the coordinate data of the standard projection. The values u_{ij} and v_{ij} are stored in the entry E_{ij} of the pano-mapping table ($E_{ij} = (u_{ij}, v_{ij})$).

Table 1: Example of a pano-mapping table of size $M \times N$. Note: $M' = M-1$ and $N' = N-1$.

	θ_0	θ_1	...	$\theta_{M'}$
φ_0	$(u_{0,0}, v_{0,0})$	$(u_{1,0}, v_{1,0})$...	$(u_{M',0}, v_{M',0})$
φ_1	$(u_{0,1}, v_{0,1})$	$(u_{1,1}, v_{1,1})$...	$(u_{M',1}, v_{M',1})$
\vdots	\vdots	\vdots	\ddots	\vdots
$\varphi_{N'}$	$(u_{0,N'}, v_{0,N'})$	$(u_{1,N'}, v_{1,N'})$...	$(u_{M',N'}, v_{M',N'})$

4. PROPOSED METHOD

PMOT is built using the data of the pano-mapping table. The values of pano-mapping table (entries E_{ij}) are not integers, therefore we interpolate its values before built the PMOT. In the proposed method, we will work with two types of interpolations (nearest neighbors and bilinear interpolation). For nearest neighbors, $f(x) = \text{round}(x)$ and for bilinear interpolation, $f(x) = \text{floor}(x)$:

$$u'_{ij} = f(u_{ij}), \quad v'_{ij} = f(v_{ij}) \quad (1)$$

The entries of the table are defined for S_{ij} . Each entry is an offset that allows obtained a specific pixel of omni-image for panoramic image generation.

$$S_{ij} = u'_{ij} + v'_{ij} \cdot L \quad (2)$$

Table 2: Example of a PMOT of size $M \times N$ ($M' = M-1$, $N' = N-1$).

	θ_0	θ_1	...	$\theta_{M'}$
φ_0	$u'_{0,0} + v'_{0,0}L$	$u'_{1,0} + v'_{1,0}L$...	$u'_{M',0} + v'_{M',0}L$
φ_1	$u'_{0,1} + v'_{0,1}L$	$u'_{1,1} + v'_{1,1}L$...	$u'_{M',1} + v'_{M',1}L$
\vdots	\vdots	\vdots	\ddots	\vdots
$\varphi_{N'}$	$u'_{0,N'} + v'_{0,N'}L$	$u'_{1,N'} + v'_{1,N'}L$...	$u'_{M',N'} + v'_{M',N'}L$

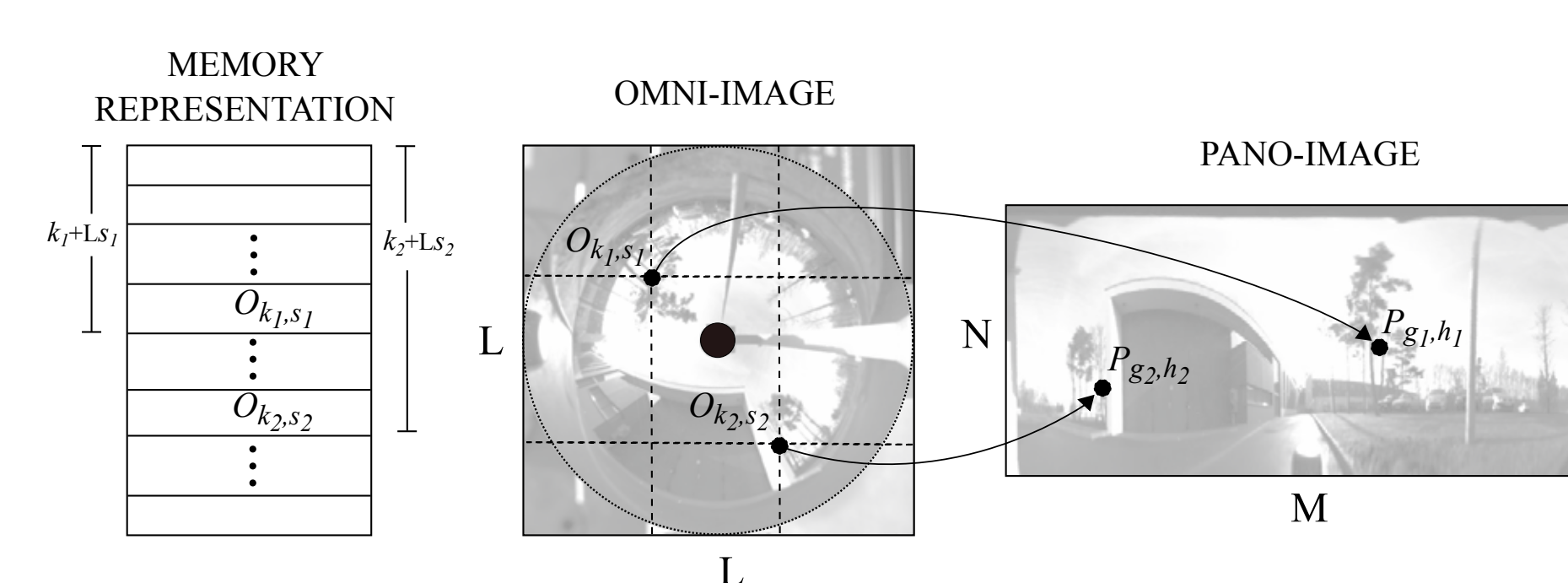


Figure 2: Offset relationship between omni-image and pano-image.

In this final step we use the PMOT for the omni-image unwarping. The equation 3 represent the general expression of bilinear interpolation, where W_1, W_2, W_3 and W_4 are the respective weights for each entry S . It's used to represent the sparse matrix multiplication. For nearest neighbors interpolation only have a weight (equal 1) and an entry S .

$$P_{ij} = W_1 \cdot O[S1] + W_2 \cdot O[S2] + W_3 \cdot O[S3] + W_4 \cdot O[S4] \quad (3)$$

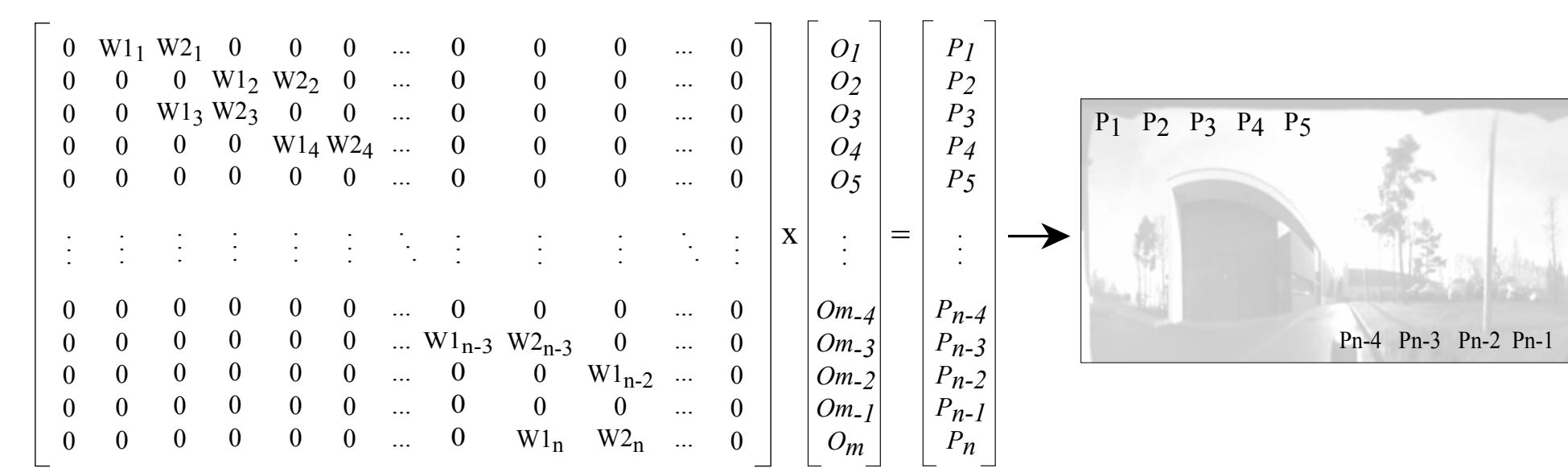


Figure 3: Sparse matrix interpretation for relationship between omni-image and pano-image. Note: $m = L \times L$ and $n = M \times N$.

We finally highlight that all the operation involved for PMOT (due to the structure that we follow) are highly parallelizable and can efficiently program on any CUDA platform.

5. COMPUTATIONAL RESULTS

We unwarped each image 100 times for the defined pano-image resolution and calculate the mean time to get an accurate estimate of the computational throughput. In the figures 4 and 5, we present our results for Jetson TK1 and Intel i5 based laptop. TK1 represents the proposed method (CUDA aware implementation), Q-core A15 (Jetson TK1) and Intel i5 represent the PMPA method (ANSI-C implementation).

Likewise, our results for nearest neighbors and bilinear interpolation show a high throughput for the TK1: 1697 ~ 34 FPS for unwarping omni-images (512x512 ~ 4096x4096) into pano-images (128x512 ~ 1024x4096). TK1 (CUDA aware implementation) has a comparable performance to Intel i5 for unwarping omni-images into pano-images (512x2048 ~ 1024x4096).

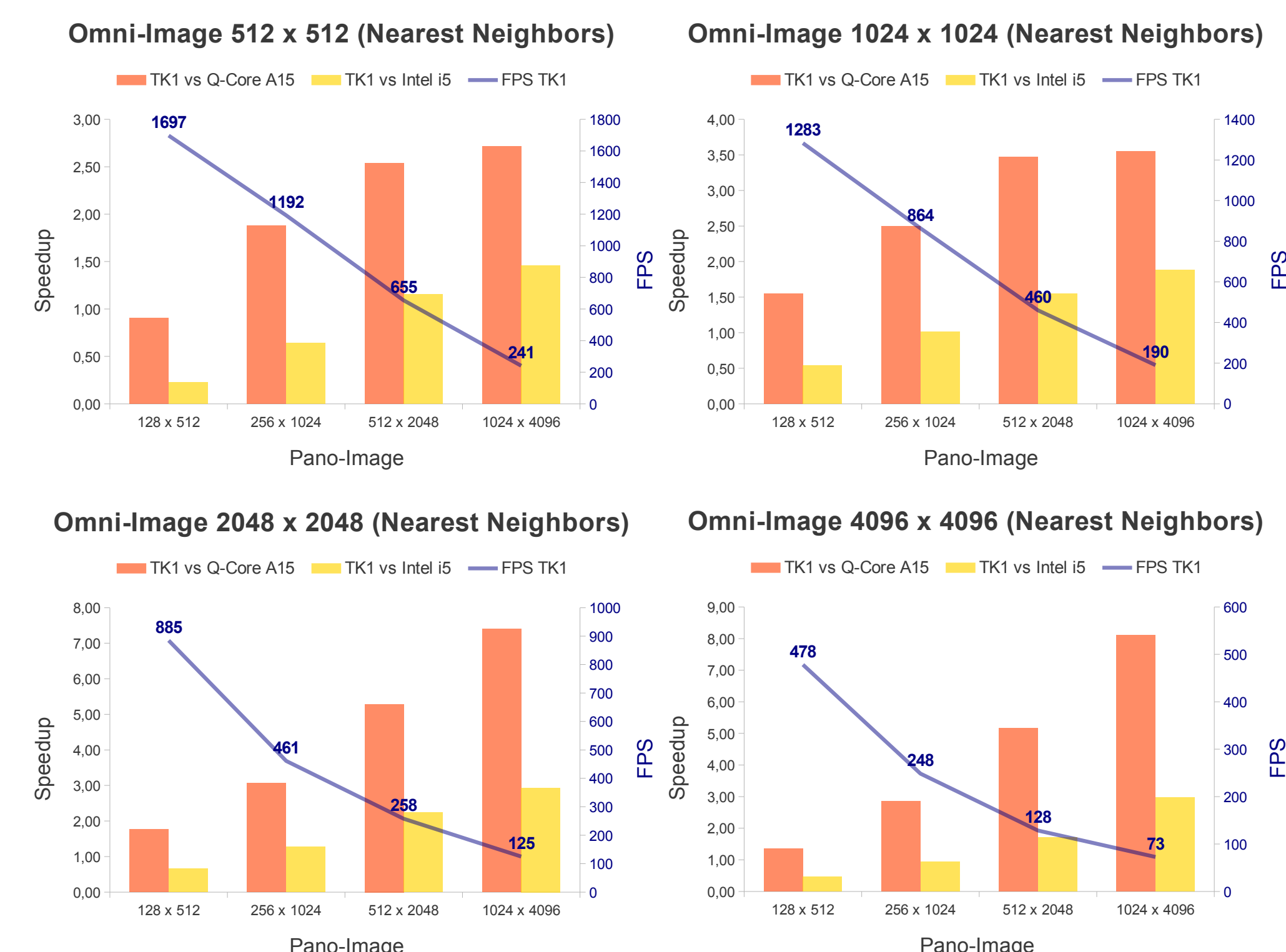


Figure 4: Results for nearest neighbors interpolation



Figure 5: Results for bilinear interpolation

6. CONCLUSIONS

- PMOT is proposed as an efficient and highly parallelizable alternative on the Jetson TK1 platform. The results shows that real-time is attained because that the proposed method can achieve more than 30 FPS even for videos with frame size of 1024x4096. Likewise, experimental results show that the proposed method outperforms processing times when compared to the state of the art.

- In the future investigations, we will study the possibility of improving the performance of omni-image unwarping using geometric symmetry relationships on CUDA platforms.

REFERENCES

- G. Wolberg and S. Zokai, "Robust image registration using log-polar transform," *International Conference on Image Processing*, vol. 1, pp. 493–496, 2000.
- A. Torii and A. Imiya, "Panoramic image transform of omnidirectional images using discrete geometry techniques," *2nd International Symposium on 3D Data Processing, Visualization and Transmission*, pp. 608–615, 2004.
- S. Jeng and W. Tsai, "Using pano-mapping tables for unwarping of omni-images into panoramic and perspective-view images," *IET Image Processing*, vol. 1, pp. 149–155, 2007.
- Z. Xiong, I. Cheng, A. Basu, W. Wang, W. Xu, and M. Zhang, "Efficient omni-image unwarping using geometric symmetry," *Machine Vision and Applications*, vol. 23, pp. 725–737, 2012.
- J. Reátegui, P. Rodríguez, and N. Ragot, "Fast omni-image unwarping using pano-mapping pointers array," accepted, 2014 IEEE Int'l. Conf. on Image Proc. (ICIP), 2014.
- N. Chong, Y. Kho, and M. Wong, "A closed form unwarping method for a spherical omnidirectional view sensor," *EURASIP Journal on Image and Video Processing*, p. 5, 2013.
- C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," *IEEE International Conference on Robotics and Automation*, pp. 3945–4950, 2007.

This research was supported by the "Fondo para la Innovación, Ciencia y Tecnología" (FINCyT) program.