Open Configuration Portable X-ray Computed Tomography

Joseph T Case[1], Shant Kenderian[1], Eric Johnson[1]

[1]Material Physics and NDE, The Aerospace Corporation, USA

X-ray computed tomography (CT) is the standard for volumetric inspection. However, it suffers from the limitation that the part or specimen under inspection is typically placed into a stationary cabinet or ring for which the x-ray source and detector are rotated. Thousands of x-ray shots are acquired at unique angles of incidence, which are subsequently processed to reconstruct a volume image of the specimen. The volume is then rendered or sliced for analysis. Obviously, not all parts are conducive to be placed in a cabinet or ring. Furthermore, it may be desired to inspect only a small region of interest on large parts. For such parts encountered in the field, only a limited number of unique angles of incidence are reasonable or possible when acquiring x-ray shots. Improved x-ray shot acquisition and reconstruction processes are desired to be able to reconstruct volume images from limited shots. Such a system may be lightweight and flexible in order to support an open configuration where arbitrary positioning of the x-ray source and detector are possible. Unfortunately, such an acquisition system may encounter positional and orientation inaccuracy. Additionally, the arbitrary placement of x-ray source and detector changes the relative distance and, consequently, the grayscale saturation. Therefore, the data acquisition and reconstruction process should take these into account. Lastly, the reconstruction process should be available in the field; that is, computation resources must be small and lightweight. Thus, it is required that the system exploit small and lightweight high performance computing (HPC) resources like graphics processing units (GPUs) or co-processors. This work describes preliminary results into a prototype system that attempts to solve the above-mentioned problems: limited shots, lightweight x-ray source and detector positioning, and efficient use of small-footprint computational hardware.