Formal description:

Input variables:

- I_p: A 360º photograph of the room (will be generated as a previous step)
- M: A 3D mesh to render in the virtual world

Output variables:

- *$L\{l_0, \ldots, l_n\}$: A set of light sources present in the real world that will be emulated in the virtual world. Each light has a location and rotation relative to the camera; and an intensity.*

Comparison to other state-of-the-art methods:
Agusanto's and Pessoa's output are texture maps that are applied to the 3D model for baked light interaction effects, such as highlights/shadows and reflections for highly specular materials. My proposed method places the lights in the actual virtual scene with real-time interactions. Xing's method already proposes using the time and date to infer the position of the sun in an outdoors scene, but the orientation of the viewer is inferred from the angles of shadows in the image. My proposed method uses the mobile device's sensors to determine the actual position and orientation of the viewer in the real world and places the simulated sun in the virtual scene accordingly.

The proposed method is composed of two phases, one offline and one real-time. During the offline phase the user will be asked to generate a 360 degree panorama using their smartphone camera. From the generated panorama, the system will determine if the sun is the main source of light in the environment. If this is the case, the sunlight will be modeled using the smartphone compass and location to determine an accurate position of the sun with respect to the camera viewpoint and positioning a light system to simulate the sun in the virtual scene in the same corresponding position.
If the sun is not the main light source in the scene (indoors or night scene) the 360 degree panorama will be used to create light sources within the virtual scene that correspond to those detected in the real world.

Related work
Agusanto et al. proposed a similar method in their 2003 paper *Photorealistic rendering for augmented reality using environment illumination*. The main differences in the method they propose have to do mostly with the technologies used. Being already an old paper, their method was conceived for Augmented Reality on a computer, rather than on a smartphone. My choice of a smartphone comes with the downside of lower processing power, but with the great advantage of the different sensors available on the device. The method in the Agusanto paper created the lightmaps using hand-crafted HDR photographs of the real scene using highly reflective metal spheres in contrast with my proposition of using the 360 degree photograph on the device itself.
Pessoa et al. expanded on Agusanto's work and created the *Real-Time Photorealistic Rendering of Synthetic Objects into Real Scenes (RPR-SORS)* toolkit as an extension of the ARToolkit Augmented reality SDK. Although they use more modern techniques, such as cubemaps instead of HDR photographs, their approach still has the same differences than the Agusanto method.
There have been several methods to insert virtual objects into static photographs automatically with highly realistic results. One such example is the method by Karsh et al in their *Depth transfer: Depth extraction from video using non-parametric sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence* paper. Tese kind of methods are not suitable for real-time applications.

A general different with all the methods reviewed is that none are thought on the context of mobile devices, they are all for AR on a computer. Also none of these methods have attempted to predict the global illumination conditions using the sun position that can be inferred from the device's sensors.