

Activity Recognition from User-Annotated Acceleration Data

3. Recovering virtual shadows

Problem: each photodiode needs to separate light rays from many different LEDs in order to recover the shadow map

Time-Based Light Beacon

- Light beacon frequency is limited by sampling rate of the microcontroller.
- However: More LEDs \rightarrow more different light beacon frequencies \rightarrow smaller interval between adjacent frequencies \rightarrow higher detection error
- **Solution:** reuse light beacon frequencies over time, reduces number of frequencies required to support a number of N LEDs.
- F_b : number of available light beacon frequencies
- Each LED i in the same group transmits with different frequency $f_i \in F_b$, same set of frequencies F_b is reused across all groups
- LED groups transmit in a fixed order of beacon slots
- LEDs that are not transmitting flash at base frequency $f_{\text{base}} \notin F_b$ to avoid two LEDs flashing at the same frequency
- **Result:** LEDs can be identified uniquely by their beacon light frequency
- Only higher density needs more beacon slots (frequencies), at a given density the system can be scaled to a large area (e.g. entire room)
- Adding time slots, requires synchronization between different LEDs and photodiodes.
- **Synchronization between LEDs:** LEDs are controlled by a central microcontroller and assigned their beacon slots
- **Synchronization between LEDs and photodiodes:** add a *preamble slot* before every set of beacon slots. All LEDs flash at frequency f_p (with $f_p \notin F_b$ and $f_p \neq f_{\text{base}}$). Photodiodes can identify this unique pattern and thus the start of the following beacon slots.
- **Frequency assignment:** Based on experimental findings. Photodiode better extracts light rays of lower flashing frequencies. Lower frequencies assigned to LEDs that are further from photodiodes, these LEDs are closer to the FoV and thus less reliable \rightarrow counteracts that

- Specifically: rank LEDs in a descend order based on their average distances to photodiodes that can perceive their light rays, assign frequency from the lowest to the highest

Blockage detection

- Given the light beacons from the LEDs, the micro-controller connecting to each photodiode performs two tasks: Sampling and processing
- **Sampling:** it periodically samples the light intensity data
- **Processing:** it locates the preamble slot and thus the following beacon slots, fetches the light intensity values within each beacon slot, computes the frequency powers at the light beacon frequencies via FFT.
- Interleaving sampling and processing steps to minimize idle time and fully use microcontroller, allows to recover shadow maps at high framerate
- **Locating Preamble:** Given D data points, locate the preamble. Sliding window method is too slow, requires performing FFT $|D|$ times. Instead divide into subgroups, perform FFT over points of each subset and identify subset D_{i^*} with peak in frequency power at f_p . Preamble slot lies in proximity to i^* . Find actual value for i^* using binary search.
- Frequency power at given frequency f_i is proportional to the light intensity from LED i
- **Adaptive Blockage Detection:** Microcontroller sends extracted frequency powers at light beacon frequencies for all beacon slots to the server. Server associates each with the respective LED based on light beacon frequency and beacon slot. Compute normalized frequency power change $\Delta P_{ij}(t)$. Light is considered blocked iff $\Delta P_{ij}(t) > \delta_{ij}$.
- Threshold δ_{ij} is adaptive, that is set based on light intensity I_{ij} and maximal light intensity I_{\max} of all light rays
- Create virtual shadow map as bitmap where bit is set iff $\Delta P_{ij}(t) > \delta_{ij}$, i.e. iff light is blocked

4. Optimizing sensor placement

Problem: finding best placement for photodiodes for accurate virtual shadow maps that best facilitate later skeleton reconstruction

- Simple bruteforce method: consider every possible combination of photodiode locations and estimate reconstruction errors \rightarrow exponential search space, not scalable
- Instead define intermediate metric to avoid involving the skeleton reconstruction algorithm
- Simple idea: light arriving at the photodiodes should be diversely spread out, maximizing

the likelihood that photodiodes capture light rays intersecting the user body (especially important for user mobility)

- For easier calculation, discretize 3D space into small uniform cubes
- **Maximization under constraint:** find placement W^* that maximizes number of cubes crossed by light rays, i.e. maximize the coverage in 3D space
- This problem is NP-hard but can be approximated in polynomial time using a greedy solution. In the experiment setup, the greedy algorithm computes optimized photodiode locations within 1 minute.

References

- [1] Li, Tianxing, Qiang Liu, and Xia Zhou. “Practical human sensing in the light.” *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 2016.