Supplementary Material

1. From Deformable to Discrete Fovea

We render SaccadeCam images with a predicted deformable attention mask when training end to end with a depth network. We outline how we transfer from deformable to discrete fovea in our paper, but provide a more detailed derivation of the optical knapsack algorithm here. We must use discrete fovea because the camera imaging our MEMS mirror has fixed spatial resolution, and we wish to cover as much of the deformable attention mask with the MEMS mirror as possible.

1.1. Optical Knapsack Derivation

Packing algorithm for varying fovea size: This problem is harder than the greedy approach because the foveal mask can change in size, which increases the number of possible combinations of selections. We cast this as a packing problem, and such theory has been studied in many domains [1] and the knapsack problem is a well-known example [3]. For us, the items in the knapsack will be mirror viewing directions $\{(\theta(V(t_1)), \phi(V(t_1))), ...(\theta(V(t_n)), \phi(V(t_n)))\}$.

We propose an attention variant on the knapsack problem that takes into account new constraints such as each mirror viewpoint's angular coverage in relation to the attention mask, reducing overlap between viewpoints and the non-uniformity of the attention mask space. Let the total size FOV available for placing mirror orientations be F, and this is determined for us by the WAC FOV. Each mirror position has its own FOV, determined by the SaccadeCam optics.

In the knapsack context, we specify weight and value of items. While the FOV is the weight of each mirror viewing direction, the value is the sum of the attention mask weights that lie within this viewing direction. We term the attention value as a_i and the FOV weight as f_i. Given n mirror viewing directions with indices $0 \leq i \leq n$, we want to find an identity vector x of length n s.t. $x_i \in (0, 1)$ and $\Sigma_i x_i a_i$ is maximized whereas $\Sigma_i x_i f_i \leq F$. While this problem is NP-hard, a pseudo-polynomial dynamic programming algorithm $O(nA)$ has been proposed by dynamic programming on an $n \times A$ array M [3].

$$
\begin{aligned}
&M[0, f] = 0 \;\; if \;\; 0 \leq f \leq F \\
&M[i, f] = -\infty \;\; if \;\; f < 0 \\
&M[i, f] = \max(M[i-1, f], a_i + M[i-1, f - f_i]),
\end{aligned}
$$

In the conventional knapsack algorithm, $M(i, f)$ always points to the largest attention value within the first i mirror vieing directions and with the FOV constraints f — and so $M(n, F)$ is the solution. For practical purposes, we can multiply these non-integers by 10^s, where s is the desired number of significant digits.

This well-known approach fails to provide the best viewing directions for SaccadeCam, because greedily increasing total attention does not guarantee *non-overlap* within the sensor FOV. In other words, a set of identical mirror viewing directions, by with consecutively increasing concentric FOVs would keep increasing the value but would redundantly cover the same angular region.

Our solution: We adapt a previous effort in computer vision for an optical knapsack algorithm [4] and present an *attention knapsack algorithm* that takes into account angular coverage by discretizing the field-of-view into β angular regions, each with a solid angle of $\frac{\pi}{\beta}$. Our key idea, inspired from [4], is to create a binary array that keeps track of the overlap of each mirror viewing direction, and the update to this does not affect the overall running time of the algorithm. We call this array $K(n, \beta)$ where $K(i, b) = 1$ if the corresponding mirror viewing direction covers this angle and is 0 if it does not.

Our method is similar to [4], and the supplementary material provides details. We also define the array M to be three-dimensional of size $n \times F \times \beta$. As before, $M(i, f, 0)$ commands the maximum attention and $M(n, F, 0)$ contains the solution. As in [4], our attention knapsack packing algorithm adds a β multiplications and $\beta + 2$ additions, still allowing a pseudo-polynomial implementation (i.e. if the number of discretizations due to β is reasonable. Please see the supplementary for the full derivation.

This results in a $O(nA\beta)$ algorithm, which is still pseudo-polynomial. As with the original knapsack problem, if the discretization of F and the angular regions β are reasonable, the implementation is tractable. We define an array $K(n, \beta)$, where $K(i, b) = 1$ if that optical element covers the angular regions b in its field-of-view, and is zero everywhere else. We also define the array M to be three-dimensional of size $n \times F \times \beta$. As before, each entry of $M(i, f, 0)$ contains the maximum attention that can be obtained with the first i viewpoints of FOV a and $M(n, F, 0)$ contains the solution to the knapsack problem. Entries $M(i, f, 1)$ through $M(i, f, \beta)$ are binary, and contain a 1 if that angular region is covered by the elements corresponding to the maximum field-of-view $M(i, f, 0)$ and a zero otherwise. The array M is initialized as,

$$M[i, f, b] = 0, \;\; \text{if} \;\; 0 \leq f \leq F, \;\; 0 \leq i \leq n \;\; \text{and} \;\; 0 \leq b \leq \beta$$

and is recursively updated as

If $f < 0$	$M[i, f, 0] = -\infty$
For any other f, **for any** i	
If $M[i-1, f, 0] < a_i + M[i-1, f-f_i, 0]$ **and** $\sum_{1 \leq b \leq \beta} M[i-1, f, b] < \sum_{1 \leq b \leq \beta} M[i-1, f-f_i, b] \vee K[i, b]$	$\begin{cases} M[i, f, 0] = a_i + M[i-1, f-f_i, 0] \\ M[i, f, b] = M[i-1, f-f_i, b] \vee K[i, b],\ b \in (1, \beta) \end{cases}$
Otherwise $\forall b$	$M[i, f, b] = M[i-1, f, b]$

where $\vee$ represents the logical OR function. This attention knapsack packing algorithm adds a β multiplications and $\beta + 2$ additions to the computational cost of the algorithm. This results in a $O(nA\beta)$ algorithm, which is still pseudo-polynomial. As with the original knapsack problem, if the discretization of F and the angular regions β are reasonable, the implementation is tractable.

2. Training and Testing Details

We use similar architectures to monodepth2 for our depth and attention networks in PyTorch. We also use the same official Eigen data split for training, validation, and test of monodepth2 [2].

We train our equiangular models for 20 epochs with a 1e-4 learning rate and 12 batch size. We train our foveated depth models with the same hyperparameters as our equiangular models, but with the same or less number of epochs as the equiangular models based on validation overfit. We also initialize both foveated and equiangular depth models with equivalent ImageNet parameters. We believe these measures ensure fair comparison between our foveated methods and equiangular methods.

References

[1] H. Dyckoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 1990.

[2] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019.

[3] S. Martello and P. Toth. Knapsack problems. *Wiley*, 1990.

[4] Francesco Pittaluga and Sanjeev J Koppal. Privacy preserving optics for miniature vision sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 314–324, 2015.