ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Class-Agnostic Robotic Gaze Control via Fast Normalized Cut

1st Andrej Lúčny
Department of Applied Informatics
Comenius University
Bratislava, Slovakia
lucny@fmph.uniba.sk

2nd Branislav Zigo
Department of Applied Informatics
Comenius University
Bratislava, Slovakia
branislav.zigo@fmph.uniba.sk

3rd Igor Farkaš
Department of Applied Informatics
Comenius University
Bratislava, Slovakia
igor.farkas@fmph.uniba.sk

***Abstract*—We present an application of a new algorithm for estimating the minimal normalized cut to the control of robotic gaze. We recursively apply the bipartition of the feature map provided by a foundation model, measuring when to stop and return object masks. We find this approach useful, stable, and capable of running in real time.**

***Index Terms*—class-agnostic, bipartition, robotics, control, normalized cut**

I. Introduction

In this paper, we introduce a class-agnostic method for turning an image into masks of the presented objects and for controlling the robot's gaze on them. This way, we investigate whether the recently developed fast normalized cut algorithm is useful for robotics applications.

II. Related Works

A. Normalized Cut

Shi and Malik [1] proposed an agnostic method for bipartitioning images using a similarity measure $w(p,q)$ among the pixels $p,q \in V$ concerning their intensity and position, such that:

- low w means different, highest w means the same
- $w(p,q) > 0$, typically $w(p,q) \in (0,1]$
- w is a positive semidefinite kernel;
  thus also $w(p,q) = w(q,p)$

Upon w, they defined measure $NCut(A,B)$ for bipartitioning of pixels V into A and B ($A \cup B = V$, $A \cap B = \emptyset$):

$$NCut(A,B) = \frac{sim(A,B)}{sim(A,V)} + \frac{sim(A,B)}{sim(B,V)}$$
$$\text{where} \quad sim(P,Q) = \sum_{p \in P, q \in Q} w(p,q)$$

is a total similarity between two sets of pixels P and Q.

They sought such bipartitioning of V into A and B for which $NCut(A,B)$ is minimal. They find that it is NP-hard, but have designed an approximate algorithm. Let $W_{p,q} = w(p,q)$ be a matrix of dimensions $n \times n$ (where $n = hw$ and $w \times h$ is the resolution of the image) expressing the similarity between every two pixels. Let $d_p = \sum_q W_{p,q}$ denote the sum of the similarities of pixel p to all others ($d = W1_n$, where 1_n is the vector of n ones), and $D = \mathrm{diag}(d)$ be a diagonal matrix such that $D_{p,p} = d_p$. They expressed a bipartition in the form of y, such that:

$$y_p = \begin{cases} 1 & \text{when } p \in A \\ -b & \text{when } p \in B \end{cases} \qquad \text{where} \quad b = \frac{sim(A,V)}{sim(B,V)}$$

and $y^T D 1_n = 0$. Then $NCut(A,B) = \dfrac{y^T(D-W)y}{y^T D y}$

As a result, having y and λ such that $(D-W)y = \lambda D y$, $NCut(A,B) = \lambda$. Thus, if λ is small, $NCut(A,B)$ approximates the minimum.

They obtained λ as the second smallest eigenvalue of the positive semidefinite matrix $R = D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}$ and the bipartition as $y > 0$ where $y = D^{-\frac{1}{2}}z$ and z is the corresponding eigenvector.

Though they can find z relatively effectively, they consume space and time quadratically with respect to the number of pixels n since they need to materialize the matrices W and R of the size $n \times n$.

B. Bipatitioning Feature Maps via Normalized Cut

Wang [2] was the first to use Normalized Cut to bipartition the feature maps produced by deep learning models into object masks. He defined $W = T_\tau(FF^T)$ where F is the matrix of size $n \times m$, where n is the number of patches (regions on the image), and m is the number of features.

C. Bipatitioning Feature Maps via Fast Normalized Cut

Lucny [3] designed the algorithm that can calculate the bipartition y without materialization of W, defining $W = FF^T$, supposing such F that the corresponding R is well-defined and positive semidefinite. The method is based on finding the eigenvector corresponding to the largest eigenvalue of

$$U = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} - \frac{z_0}{|z_0|}\left(\frac{z_0}{|z_0|}\right)^T$$
$$\text{where} \quad z_0 = D^{\frac{1}{2}} 1_n$$

It is calculated by the power iteration using U for which

$$Uy = unproj(y, (d^{-\frac{1}{2}} 1_m^T) \circ F) - unproj(y, \frac{z_0}{|z_0|})$$
$$\text{where} \quad unproj(u, v) := v(v^T u)$$

This approach scales linearly with the number of patches (or pixels). Moreover, the bipartition $y > 0$ converges much faster than the eigenvalue, so the number of necessary iterations is exceptionally low (1, 2, 4, or 8, depending on the resolution and the number of features).

III. Method and Results

A. Recursive Fast Normalized Cut

When we run a foundation model on an image (we have used DINOv3 [4]), we get a feature map that describes the content of the image patches (384 features for 48×48 patches). Yet we can interpolate this feature map to a much finer resolution, e.g., 320×240 of the original resolution of the robot's eye camera. Then, since our bipartitioning algorithm is fast, we can run it recursively to split the image into a system of areas containing similar features (Figure 1). The problem is to determine whether we already have an object and avoid continuing its bipartitioning. The best results we have got with the rule that a component is split only if the cosine similarity between the average features of the two parts proposed by the bipartitioning algorithm meets:

$$\frac{\overline{F[A]} \cdot \overline{F[B]}}{|\overline{F[A]}||\overline{F[B]}|} < 1 - \tau d$$

where τ is a threshold (we used 0.05) and d is depth of the bipartition $(0, 1, 2, \dots)$

B. Gaze Control

Achieving 6 fps on a gaming notebook, the control of a robot's head following objects we present is straightforward (Figure 2). The robot reacts to the presence of a complete object at the center of its field of view. We found that the algorithm is even too stable, and it would be convenient to add occasional random head movements to make it more human-like when no objects are present, or they are present but fixed. The contribution of this approach is that we can use really any object. Moreover, we obtain a quite fine object mask and a descriptor that enables us to distinguish among the objects.

Fig. 1: Examples of the results of the recursive normalized cut.

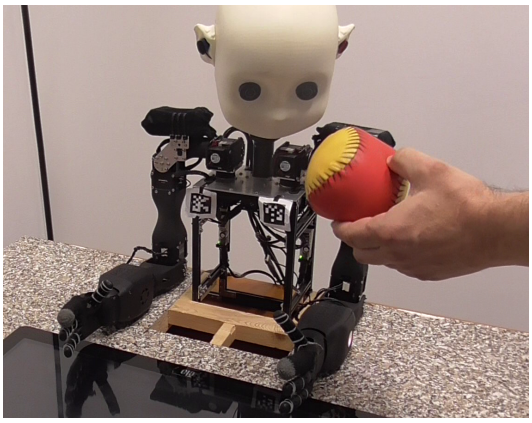

Fig. 2: Robot NICO's gaze controlled via the recursive normalized cut.

IV. Conclusion

We have shown that the Fast Normalized Cut algorithm can enable new interesting applications in robotics and elsewhere. Code is available at https://github.com/andylucny/fastncut.git

Acknowledgment

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I01-03-V04-00048 (HUROSI).

References

[1] Jianbo Shi, Jitendra Malik (2000). *Normalized cuts and image segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 888–905.

[2] Xudong Wang, Rohit Girdhar, Stella X. Yu, Ishan Misra (2023). *Cut and Learn for Unsupervised Object Detection and Instance Segmentation (CutLER).* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023) (pp. 3124–3134).

[3] Lúčny, A. (2026). A Fast Algorithm for Normalized Cut with Applications on Bipartitioning Feature Maps in Deep Learning. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.6482332

[4] Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., … Bojanowski, P. (2025). DINOv3: Self-supervised learning for vision at unprecedented scale. arXiv. https://doi.org/10.48550/arXiv.2508.10104