Biternion Nets: Continuous Head Pose Regression from Discrete Training Labels

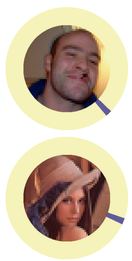
Lucas Beyer, Alexander Hermans, Bastian Leibe

1. The Main Idea

Inferring the orientation of people's heads is an important low-level task for scene-understanding and robot navigation. Most current approaches have either of the following drawbacks:

- They can't deal with the periodicity of angular data. These methods are usually evaluated only on front-facing datasets.
- They require *very* fine-grained regression labels. These are rare, usually only front-facing, and tedious to create.

We show how CNNs and a biternion encoding solve both of these issues. We obtain state-of-the-art results on several known datasets and show that we can learn a continuous output from discrete annotations.



2. A Good Architecture

The network architecture is based on the popular VGG one. By itself, perhaps unsurprisingly, it beats all other state-of-the-art methods on currently available head-orientation benchmarks for classification

	HIIT	HOCoffee	HOC	QMUL	
# Samples	12 000/12 007	9522/8595	6860/5021	7603/7618	9813/8725
# Classes	6	6	4	4	4+1
Tosato et al.	96.5%	81.0%	78.69%	94.25%	91.18%
Lallemand et al.	-	-	79.9%	-	-
Our CNN	**98.70%**	**86.99%**	**83.97%**	**95.58%**	**94.30%**

and regression.

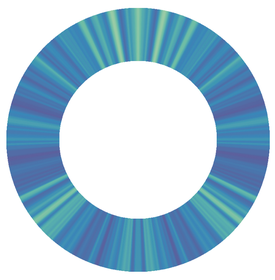
	IDIAP Head Pose			CAVIAR-c	CAVIAR-o
# Samples	42 304/23 991			10 660/10 665	10 802/10 889
Pose Range	pan	tilt	roll	pan	pan
	[-101,101]	[-73,23]	[-46,65]	[0, 360]	[0, 360]
Tosato et al.	10.3° ± 10.6°	4.5° ± 5.3°	4.3° ± 3.8°	22.7° ± 18.4°	35.3° ± 24.6°
Ba & Odobez	8.7° ± 9.1°	19.1° ± 15.4°	9.7° ± 7.1°	-	-
Our CNN	**5.9° ± 7.2°**	**2.8° ± 2.6°**	**3.5° ± 3.9°**	**19.2° ± 24.2°**	**25.2° ± 26.4°**

But these datasets are not interesting because none of them covers the case of full 360° orientation regression.

(CAVIAR's label distribution is almost exclusively concentrated around the four canonical orientations.)

3. Periodic 360° Regression

There is a discontinuity at the 0°/360° limit: a prediction of 372° is equally good or bad as a prediction of 12°.



To investigate this problem, we turn to the Town Centre dataset, which has full 360° annotations and a more-or-less uniform label distribution as shown to the right. Roughly, we've gone through the following thinking:

1. So what? Just ignore the problem!
   Result: **38.9°**±40.7° prediction error. (A shallow network has 64.1°±45.0° error.)
2. Put a modulo on the cost function. This introduces a huge jump into the function, which confuses gradient-based methods.
   Result: **divergence**.
3. Use the von-Mises distribution. It's like a Gaussian on a circle.
   Result: **29.4°**±31.3° prediction error.

$$C_{\mathrm{VM}}(\varphi \mid t; \kappa) = 1 - e^{\kappa(\cos(\varphi - t) - 1)}$$

4. Biternion Representation

But there's still a problem: the network is forced to a linear output, while an orientation really is a circular value.

We can represent an angle through its sine and cosine, i.e. as a point on the 2D unit circle: $\mathbf{y} = (\cos\varphi, \sin\varphi)$. We call this the biternion representation due to its relation to quaternions commonly used for rotations in 3D graphics.

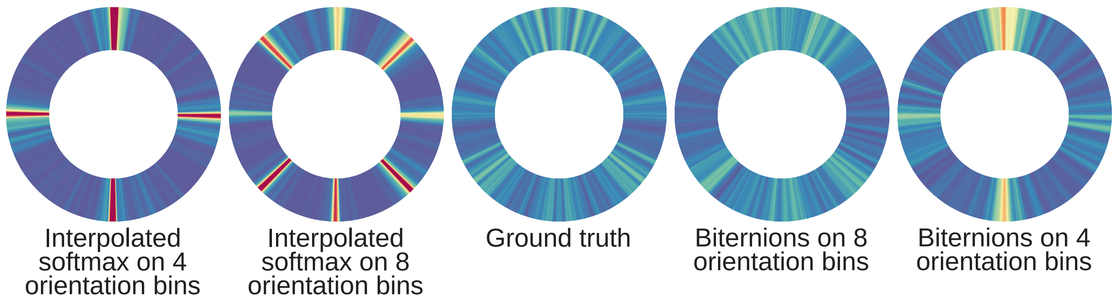
This representation can be encoded into the network by using a two-dimensional output followed by a normalizing nonlinearity. Such an output naturally leads to using the cosine cost-function.

4. So let's make the output layer circular!
   Result: **21.6°**±25.2° prediction error. (And **20.8°**±24.7° using von-Mises cost.)

5. From Discrete to Continuous

Creating a dataset of head-orientations with precise, continuous annotations is a tremendous amount of hard work. Is there a way to make continuous predictions without requiring such effort?

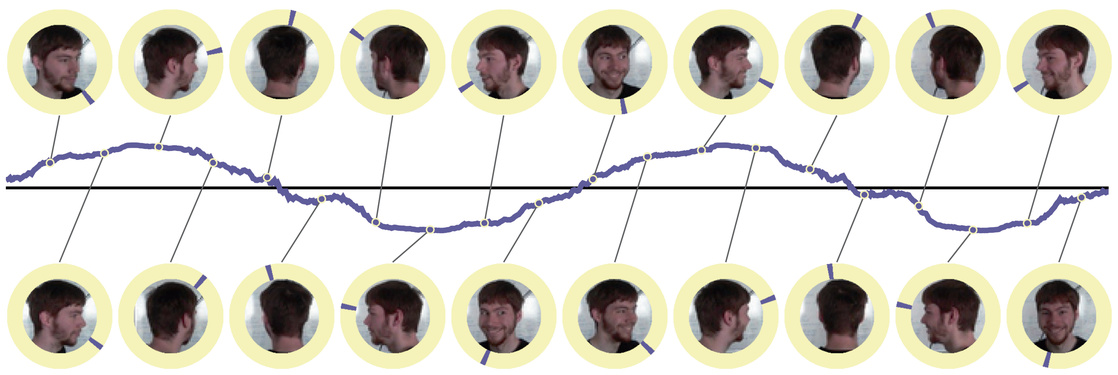
Because CNNs are a continuous input-output mapping and SGD optimization just "pulls" on the mapping function, we believe it is capable of learning a sensible, continuous orientation predictor on roughly-labeled data, e.g. *left*, *right*, *front*, *back*. To investigate, we discretize TownCentre's labels and get the following results.



Interpolated softmax on 4 orientation bins | Interpolated softmax on 8 orientation bins | Ground truth | Biternions on 8 orientation bins | Biternions on 4 orientation bins

6. Practicality

To demonstrate that our method can actually be used in real-life scenarios within a reasonable development timeframe, we proceeded to record videos of 24 labmates turning on the spot in front of various backgrounds. A single person cropped and annotated the dumped images into eight bins in no more than 2h30. This is what a BiternionNet trained on that dataset predicts for an unseen person, frame-by-frame:



The blue curve shows the sine of the angle inferred by the network. As is clearly visible, the output is very smooth, continuous, and there is no outlier.

The full paper has been published at GCPR'15. Code is available:

`http://github.com/lucasb-eyer/BiternionNet`

Visual Computing Institute
Computer Vision
Prof. Dr. Bastian Leibe

RWTH AACHEN UNIVERSITY