Half Hypersphere Confinement for Piecewise Linear Regression

Eduardo Pérez-Pellitero[1,2], Jordi Salvador[2], Javier Ruiz-Hidalgo[3] and Bodo Rosenhahn[1]

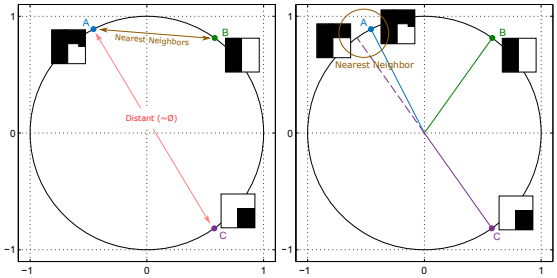
[1]TNT Lab, Leibniz Universität Hannover
[2]Technicolor R&I Hannover
[3]Image Processing Group, Universitat Politècnica de Catalunya

Abstract

Recent research in piecewise linear regression for Super-Resolution has shown the positive impact of training regressors with densely populated clusters whose datapoints are tight in the Euclidean space. In this paper we further research how to improve the locality condition during the training of regressors and how to better select them during testing time. We study the characteristics of the metrics best suited for the piecewise regression algorithms, in which comparisons are usually made between normalized vectors that lie on the unitary hypersphere. Even though Euclidean distance has been widely used for this purpose, it is suboptimal since it does not handle antipodal points (i.e. diametrically opposite points) properly, as vectors with same module and angle but opposite directions are, for linear regression purposes, identical. Therefore, we propose the usage of antipodally invariant metrics and introduce the Half Hypersphere Confinement (HHC), a fast alternative to Multidimensional Scaling (MDS) that allows to map antipodally invariant distances in the Euclidean space with very little approximation error. By doing so, we enable the usage of fast search structures based on Euclidean distances without undermining their speed gains with complex distance transformations. The performance of our method, which we named HHC Regression (HHCR), applied to Super-Resolution (SR) improves both in quality (PSNR) and it is faster than any other state-of-the-art method. Additionally, under an application-agnostic interpretation of our regression framework, we also test our algorithm for denoising and depth upscaling with promising results.

1. Introduction

Super-Resolution (SR) techniques aim to recover a high-resolution (HR) image from a low-resolution (LR) input image. In order to achieve the desired HR image we must determine the value of more variables than observations are available, thus the problem is ill-posed. The common approach in the literature is to regularize the problem by the addition of natural image priors: a great variety of them have been presented across the years, yet there is an ongoing shift towards *learned* priors.



(a) Euclidean metric (b) Antipodally invariant metric

Figure 1: Behaviour of Euclidean distance and angular distance for points A, B and C. Although A and C have very similar structures, Euclidean distance fails to group them together.

The earlier priors based on more or less rigid models of the imaging process (e.g. the bicubic smooth prior, the Iterative Back Projection (IPB) of Irani and Peleg [12]) have been outperformed by priors which use learning or statistical modeling to regularize the problem from training examples, usually at a patch level (e.g. the opening work of Freeman et al. [8], the cross-scale self-similarity prior of Glasner et al. [9] recently refined to extend the search domain via transformed patches [11], gradient priors such as [19] or those using deep learning [6, 3]). An important and prolific prior was that of sparsity, which was widely used not only in SR [24], but also in many other image enhancement problems [13, 7]. This approach assumed the decomposition of a natural patch to be a sparse subset of entries from a compact dictionary, and its main drawback is still

the computational cost of the L_1 norm.

As a solution to the problematic of online sparse patch decomposition, Timofte et al. [20] introduced a discrete set of basis for which a neighbor embedding was calculated offline, thus avoiding an overhead during testing time. In addition to that, they proposed substituting norm L_1 by L_2 as a norm relaxation, obtaining very competitive results. This SR method, named Anchored Neighborhood Regression (ANR) is composed by a set of anchored regressors and the SR upscaling is done through a simple matrix multiplication. Briefly afterwards, an improvement for ANR was presented supporting the usage of densely populated neighborhoods (i.e. clusters) of samples very close in the space, which respects better the L_2 norm collaborative representation nature [21, 15]. Within the family of regression algorithms, several works introduced the usage of sublinear search structures such as trees [23, 22], hashing [15] or even ensemble of trees (i.e. forest) as the recent work of Schulter et al. [18].

In this paper we follow-up on the direction of ANR, using an ensemble of piecewise linear regressors trained offline, a regressor selection stage (done through a sublinear search structure, as in [15]) and a matrix multiplication as the only steps of the SR algorithm. In our algorithm, we introduce distance metrics other than Euclidean distance (angular metrics), under the motivation that Euclidean distance fails to group antipodal points (i.e. diametrically opposite points) as nearest neighbors. We also design a transformation in order to preserve the relative angular distances in the Euclidean space and, therefore, enable the usage of any sublinear search strategy. To conclude, we interpret our framework as an application-agnostic regression which adapts from a coarse first estimator and is able to address other problems of the same nature (e.g. denoising). As to performance, our proposed HHCR-SR method surpass the most recent state-of-the-art by 0.2 to 0.3dB and it does it about $\times 4 - 11$ times faster.

2. Proposed method

The proposed method is based on piecewise linear regression, which we denote as HHCR and which builds on the recent work of [21]. It consists in an ensemble of linear regressors anchored to certain centroids and performs the regressor search through a Spherical Hashing (SpH) algorithm, as the regression ensemble in [15]. We train and select the regressors differently, as shown in the upcoming sections, as well as introduce a mapping transformation of both the centroids and the input observations in order to optimize the sublinear search performed by the SpH.

2.1. Piecewise Linear Regression

Regression analysis aims to estimate a certain transformation between observed variables which, in a broader sense, models the relations between the manifolds in which those variables are laying. A multivariate linear regression is essentially a matrix multiplication:

$$x = R\,y, \tag{1}$$

where $y \in \mathbb{R}^m$ is an input observation, $x \in \mathbb{R}^n$ the resulting output and $R \in \mathbb{R}^{n \times m}$ is the regressor. This operation requires $n(2m-1)$ operations, i.e. the complexity is $\mathcal{O}(nm)$.

The challenges present in low-level vision are usually too complex to be addressed by a simple linear regression. The under fitting commonly present in linear regression is diminished by a multiplicity of different linear regressors that conform a piecewise linear regression system. The usage of these ensembles is and has been common in many applications, sometimes under hierarchical structures such as regression trees or regression forest (where a single linear regressor is attached to each tree leaf).

In order to select the best regressor R_i (which has an associated centroid c_i) from within the ensemble $\{R_k\}$ a certain criteria $\delta(c_i,\, y)$ needs to be defined, e.g. distance metric, tree traversal. Different regressors are applied to the input observations depending on this criteria:

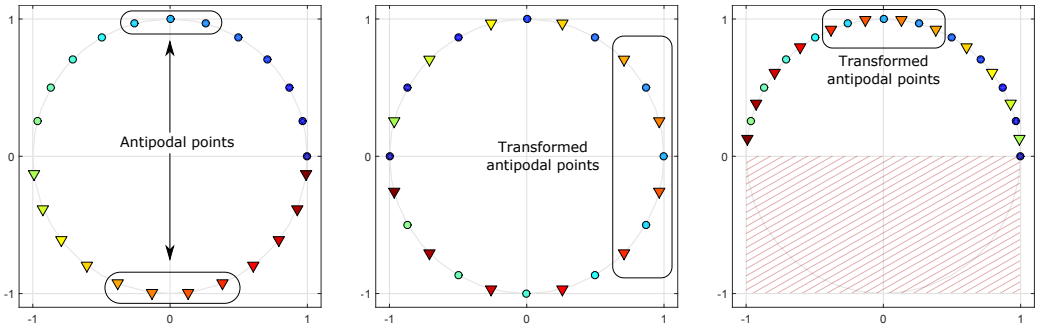
$$x = R_i\,y,\ s.t.\ R_i = \underset{c_i \in \{c_k\}}{\operatorname{argmin}}\, \delta(c_i,\, y). \tag{2}$$

Recently, some methods have applied this framework and structure to the ill-posed challenge of Super-Resolution (SR) imaging [23, 21, 20, 15], showing its great potential and performance.

2.2. Metrics for linear regression

The linear regression scheme is, as we have seen, very straightforward. One of the most fundamental aspects of the system is how we choose the best-suited regressor, i.e. the metric used to compare the input patch to the ith centroid. This metric is not only important during testing time, but also during training time to asses which observations are used to train which regressor. It is recurrent in literature the use of Euclidean distance for this purpose. If we are aiming a nearest neighbor search for a regression system, Euclidean space without any transformation is suboptimal as it is not further exploiting the intrinsic characteristics of linear regression.

The scalar matrix multiplication gives us some information about the ambiguous variations that the metric we want to define should ignore, i.e. for a given scalar λ we obtain $\lambda x = R(y\lambda)$. The regressor R and the associated linear operations are not changed by this scaling operation. Therefore, performing a vector normalization is a good practice as it solves partially the undesired variability derived from scalar multiplication. Unitary vectors collapse all positive scalar variations into a single unitary vector, thus holding more training examples available for a certain vector type



(a) Input points (b) MDS transformation (c) Proposed transformation

Figure 2: 2D example of our HHC compared to MDS. (a) Input points distributed on the unitary sphere (each point has its own color label), circles/triangles denote positive/negative y-axis coordinate. (b) Points obtained with MDS and angular distance. (c) Points obtained with our proposed fast transformation. To guide the reader there is a group of antipodally invariant nearest neighbors highlighted in a box across the three figures. Note how both MDS and our proposed transformed put close together antipodal nearest neighbors as opposed to the input data, which were located at maximum distance.

and being able to use them efficiently. During testing time, although the regression must be done with the original non-normalized vector y, the search should be done with the normalized version $\hat{y} = \frac{y}{\|y\|}$ for the same principles.

However, there are still certain cases which are not properly managed by just a normalization, as the norms are strictly positive, i.e. $\|y\| \ \epsilon \ \mathbb{R}^+$, and therefore can not compensate for all those scalar values $\lambda \ \epsilon \ \mathbb{R}^-$. In a unitary sphere composed by normalized vectors, the case of a negative λ represents its antipode (i.e. the point that is diametrically opposed in the unitary sphere).

The antipode of a point is one of the two closest possible nearest neighbor, however in the Euclidean space they are the most far away possible points (i.e. at a diameter distance) as it can be appreciated in Fig. 1. A good metric for regression should be therefore *antipodally invariant*.

We propose a metric based on the cosine similarity (CS) as a native antipodally invariant similarity metric which is well adapted for regressors' nearest neighbor search:

$$\varsigma(c,\, y) = |\hat{c} \cdot \hat{y}| = |\cos\theta|\,, \tag{3}$$

whose output is bounded in the $[0,\ 1]$ range (1 denotes maximum similarity) and measures the absolute value of the cosine of the angle θ between the two vectors c and y. The equivalent distance metric, which we denote as angular distance reads:

$$\delta_\theta(c,\, y) = \frac{2}{\Pi} \arccos(\varsigma(c,\, y)) \tag{4}$$

and is normalized to be in the range $[0, 1]$ range (1 denotes maximum distance).

When there is no time nor metric space constrains (e.g. during training), using the similarity calculation of Eq. (3) is the best option. However, during testing time if a binary split is used, and this split is making use of Euclidean space (such as the one used in [23, 15]), the adaptation is not straightforward. Rather than trying to design a split-specific metric, as in the contemporary antipodally invariant naive Bayes forest SR [17], we study the embedding of our datapoints in the Euclidean space in such a way that antipodally invariant distances (i.e. Eq. (4)) are preserved.

2.3. Embedding in the Euclidean Space

Multidimensional scaling (MDS) is a well known statistical method that transforms an $l \times l$ matrix D containing pairwise distances between all l observations into a set of coordinates such that the Euclidean distances derived from them preserve the relative distances specified in D. MDS is widely applied as a metric-preserving dimensionality reduction method, as the dimension of the output coordinates is user specified [16].

Although MDS can map appropriately antipodally invariant distances into Euclidean space, it is unusable as not only the optimization process is computationally intense when the number of points is very large (its complexity is $\mathcal{O}(ml^2)$, where m is the dimensionality), but additionally the input matrix D requires performing an exhaustive search point-to-point for all l elements. The Landmark MDS (LMDS) of Silva and Tenenbaum [5] introduced a more efficient transformation based on an approximate anchored MDS.

Silva and Tenenbaum propose to divide the algorithm in two steps: a first step in which a classical MDS is performed with a smaller set of points, i.e. landmark points $l_s \ll l$, and a second step that applies a distance-based triangulation in order to obtain the embedding of the complete l elements. The first step can be done beforehand in a training stage, selecting an optimal set of landmark points, whose minimum size is $m + 1$ landmarks for a m-dimensional embedding. The embedding vectors for each of the points can be ob-

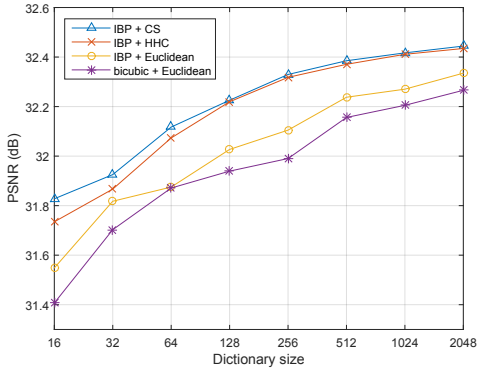


Figure 3: Super-Resolution upscaling PSNR vs dictionary size for different metrics and coarse approximations. All the configurations use exhaustive search.

tained as:

$$q_a = -\frac{1}{2} L_m^{\#}(d_a - d_\mu), \tag{5}$$

where d_a is a vector with the squared distances from point a to all the landmarks, d_μ is a vector with the mean square distances from the i-th landmark to all the landmarks (it is obtained in step 1) and $L_m^{\#}$ is the pseudoinverse transpose of L_m (it is obtained in step 1, we refer the reader to [5] for further details). LMDS only requires calculating the distances for $l_s \times l$. Nevertheless, although substantially faster than the original MDS, LMDS is still causing a big impact in the processing time in algorithms which have an emphasis in low complexity, such as it is our proposed work.

As a fast alternative to the MDS family of algorithms we propose a simple deterministic transformation designed to mimic the MDS behavior when it is used with angular distances. Fig. 2 (a) and (b) show the coordinates obtained with MDS with a D matrix constructed with angular distances. If we analyze the transformation in the y-axis, the MDS transformation stretches the positive half-space points to occupy the whole sphere, and it maps likewise the negative half space. The resulting mapping contains both original half-spaces mixed in such a way that the angular distances are preserved. The intuition behind our transformation is to make use of the inverse projection ($\lambda = -1$, which is neutral for the regressor search) to compress all the data in the positive half space rather than stretching both half-spaces, as it can be seen in Fig. 2 (c).

Several conditions need to be met for our proposed transformation to be effective. It takes advantage of the characteristics of normalized features (i.e. observations in the unitary sphere S^{m-1}). It also requires them to be distributed in both positive and negative half-spaces in a balanced way at least in one dimension.

The desired function must map two (antipodal) points in S^{m-1} into a single point. In order to do that, we enforce a forbidden space region, corresponding to the negative half-space of the qth dimension, i.e. the observations must be $y \cdot \mathbf{e_q} \in \mathbb{R}^+$, where $\mathbf{e_q}$ is the qth standard basis in the Euclidean m space:

$$y_{TR} = -y, \qquad \text{if } y \cdot \mathbf{e_q} < 0. \tag{6}$$

In our training (around 500K feature vectors), all the dimensions were highly and similarly balanced, so any of them could be chosen. Whenever this is not the case, the most balanced dimension should be selected to create the hyperplane.

The outcome is a Half Hypersphere Confinement (HHC) instead of the initial unitary hypersphere, where the Euclidean distances respect also the angular distances.

The proposed HHC is created from an hyperplane $y \cdot \mathbf{e_q} = 0$ which is the bound of the confinement. The performance of our proposed HHC transform depends on the distance to this hyperplane, as points which are very close to the hyperplane loose connection to the points immediately below the hyperplane, which are projected to the upper half hypersphere. As shown in Fig. 2, MDS has a continuous distribution of points while our proposed transform is truncated in the $y \cdot \mathbf{e_q} = 0$ hyperplane. However, in Fig. 5 we quantify the low incidence of this behavior by measuring a similarity ratio $\eta = \zeta_{HHC}/\zeta$ vs the distance to the hyperplane and observing its frequency. Although there is indeed a certain degradation for small distances to the hyperplane, the frequency is very low and most of the similarities obtained with HHC are highly reliable (99.2% of the total amount of points have a similarity ratio higher than 0.85).

Similarly to the work of [15], we place a Spherical Hashing search split [10] on top of our piecewise linear regression using the HHC to embed our points in the Euclidean space. Spherical hashing differs from previous approaches by using hyperspheres to define hashing functions on behalf of hyperplanes. A given hashing function $H(y) = (h_1(y_F), \ldots, h_s(y))$ maps points from $\mathbb{R}^m$ to a base 2 $\mathbb{N}^s$, i.e. $\{0,1\}^s$. Every hashing function $h_k(y)$ indicates whether the point y is inside kth hypersphere, modeled for this purpose as a *pivot* $p_k \in \mathbb{R}^m$ and a distance threshold (i.e. radius of the hypersphere) $t_k \in \mathbb{R}^+$ as:

$$h_k(y_F) = \begin{cases} 0 & when\ \delta(p_k, y) > t_k \\ 1 & when\ \delta(p_k, y) \leq t_k \end{cases}. \tag{7}$$

In Fig. 3 we show first the superiority of the antipodal invariant metric CS with respect to Euclidean distance in a regression-based SR application, thanks to a system where regressors are trained with closer and denser neighborhoods and the selection of regressors in testing time is improved. Additionally, we show the good performance of HHC for SR upscaling, which approximates closely the performance

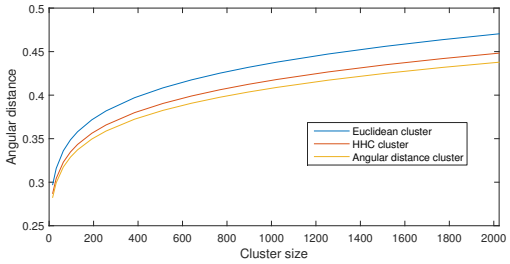


Figure 4: Average angular distances for clusters built with different metrics.

obtained with CS, specially from 128 atoms on. Note that in [15], when placing a sublinear search structure based on the Euclidean distance they introduced a substantial quality drop with respect to exhaustive search. Differently, in our proposed scheme of SpH together with our proposed HHC, the drop in quality is very reduced or even nonexistent (see Table 1).

2.4. Training

The training of regressors is performed from certain centroids obtained through any unsupervised clustering technique, such as spherical k-means [14] or K-SVD [25], in order to obtain a set of k centroids $\{c_k\}$. As the previous original work of Zeyde et al. and Timofte et al. [25, 20], we also found in K-SVD a good match for this first clustering stage. It is important to note that K-SVD is well adapted to work with antipodal points, so it fits the scope of this paper. From this moment on, we build each cluster C through a NN search against a large pool of training samples $\{y\}$ from which we know the ground truth $\{x\}$. As we are not time constrained, we use angular distance for our search. As it can be seen in Fig. 4, clusters created with both angular distance and HHC obtain smaller distances than Euclidean clusters, which is optimal for the regressors training [15, 21] as either the observations are closer to the centroid at a given cluster size, or for a given maximum distance there are more available training samples.

3. Applications

The regression scheme presented in this paper is based on the original SR work of ANR from Timofte et al. [20]. Although not explicitly stated in the paper, they divide their output x into two separate components, i.e. $x = \tilde{x} + x_R$, which divides the signal into two components: those obtained through regression x_R and the less challenging ones $\tilde{x}$, that are obtained otherwise. The strategy is to carry out the estimation $\tilde{x}$ with a low-complexity well-known estimator, e.g. bicubic for SR applications, $\tilde{X} = est(Y)$. The resulting regression scheme reads:

$$x = \tilde{x} + Ry. \tag{8}$$

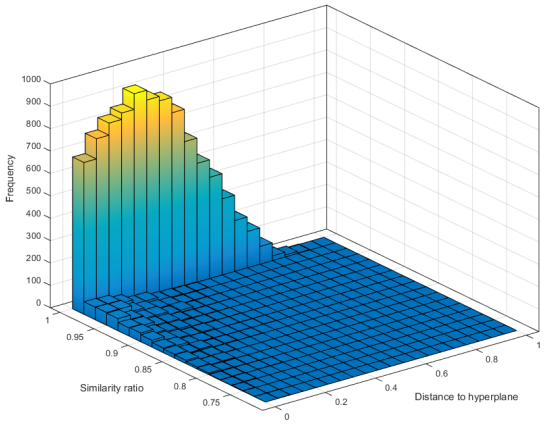


Figure 5: Histogram of the similarity ratio vs distance to the folding hyperplane. We evaluate the well-functioning of our proposed transformation by searching the 1-st nearest neighbor from 10k points to 1024 centroids (i.e. the testing case of our regression ensemble). We first obtain the NN both with cosine similarity (i.e. best solution) and with our proposed transformation of the Euclidean space (i.e. approximation). We recalculate the cosine similarity for both NN and compute the ratio $\eta = \zeta_{HHC}/\zeta$ (when $\eta \approx 1$ the approximation is very close to the best solution).

This strategy present several advantages: The regression performance improves when it starts from a better approximation, so the better the estimator used, the better the final result. Secondly, the subtraction $x_R = x - \tilde{x}$ has zero mean value and thirdly, this scheme makes the regression ensemble application-agnostic. By only changing this coarse estimation the regression framework adapts to the new characteristics of the application. In this paper we show the performance of our algorithm for Super-Resolution (not only luminance but also depth images) and Denoising.

3.1. Super-Resolution

Super-Resolution results can be obtained by using a simple upscaler to estimate $\tilde{X}$, e.g. bicubic. Several research has dealt with similar regression frameworks, demonstrating that piecewise linear regression is very well suited to minimize the blur and aliasing artifacts introduced by other simple upscalers. In our testing we validate that antipodally invariant metrics, performed through our HHCR in the Euclidean space, widely outperforms the state-of-the-art SR methods. We also introduce an additional improvement by extracting gradient features over an upscaled image $\tilde{X}$ obtained by a simplified Iterative Back Projection (IBP) [12] algorithm rather than the usual bicubic. In Fig. 3 we show the difference between bicubic and IBP-based gradi-

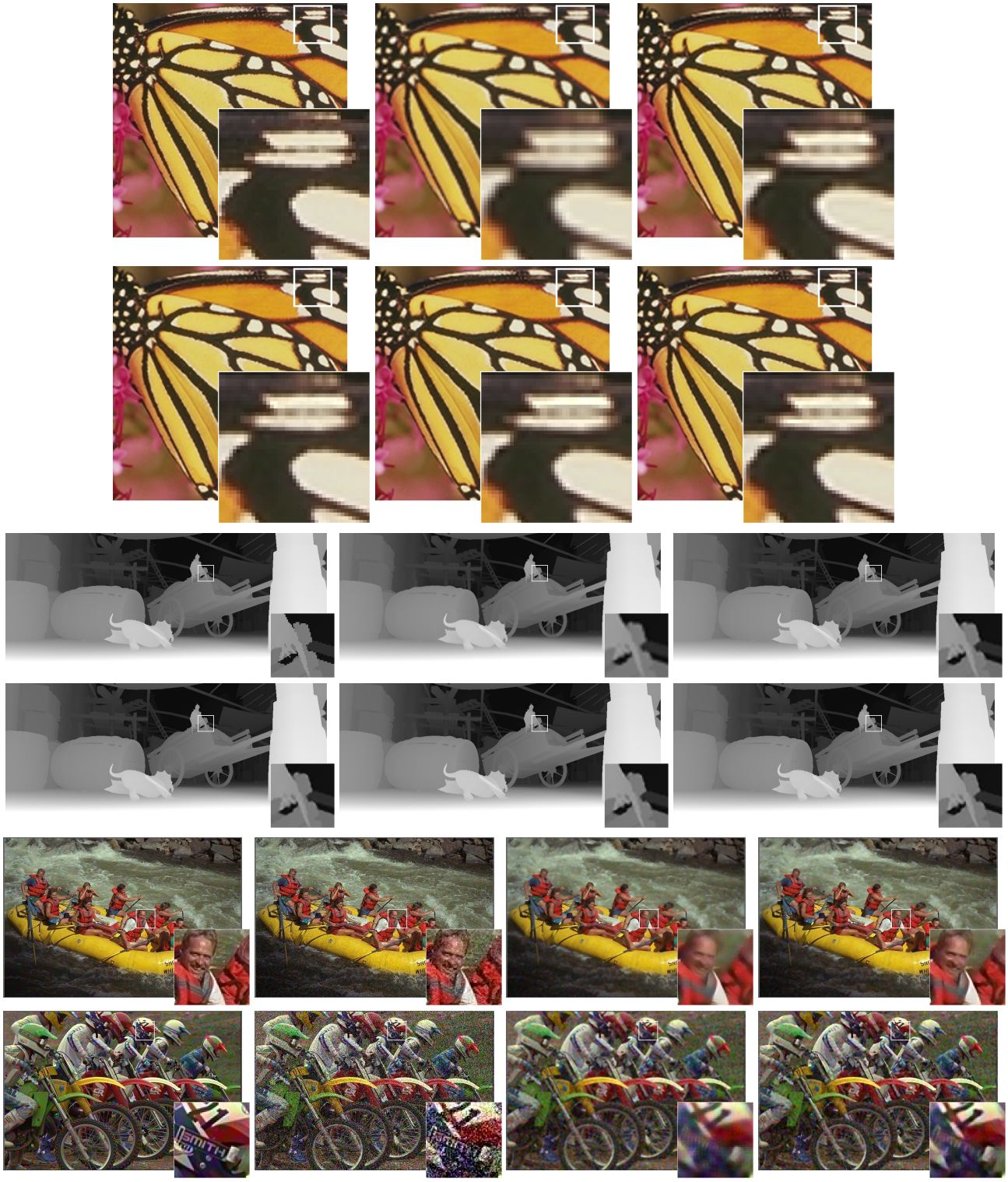


Figure 6: Visual qualitative assessment for several applications and datasets. The *butterfly* image (Set5) and the *market* depth image (Market) are upscaled with a $\times 2$ magnification factor. From left to right and top to bottom: Original, bicubic interpolation, ANR [20], ASRF [18], A+ [20] and HHCR with 7 hyperspheres. The last two groups of images (*boat* and *motorbikes*) show denoising in the kodak dataset, from left to right: Original, Noisy, NLM, NLM + HHCR (first row $\sigma = 0.05$, second row $\sigma = 0.15$). Better viewed zoomed in.

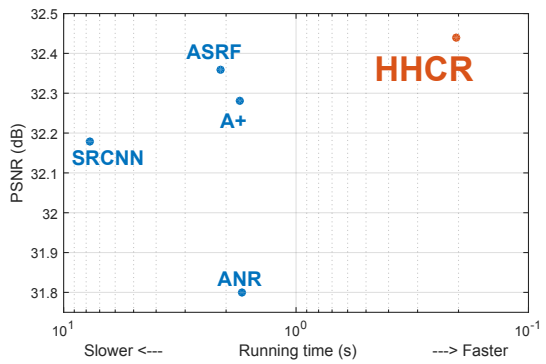


Figure 7: Our proposed HHCR-SR achieves both the best quality (PSNR) and the fastest speed (s) in comparison with most recent state-of-the-art methods. All algorithms run Set14 with a $\times 2$ upscaling factor, see Table 1 for more information.

	σ	Noisy PSNR	NLM[1] PSNR	NLM[1] Time	NLM+HHCR PSNR	NLM+HHCR Time	BM3D[4] PSNR	BM3D[4] Time	BM3D+HHCR PSNR	BM3D+HHCR Time
Kodak	0.05	26.03	29.98	**0.204**	31.59	0.932	33.64	6.400	**33.70**	8.994
	0.10	20.11	27.19	**0.199**	28.70	0.955	30.47	6.496	**30.53**	9.303
	0.15	16.75	25.85	**0.204**	26.93	0.989	28.75	6.421	**28.80**	9.472
	0.20	14.50	24.87	**0.209**	25.55	1.000	27.63	7.573	**27.68**	9.365
Set5	0.05	26.08	31.80	**0.064**	31.21	0.303	34.39	1.623	**34.52**	2.475
	0.10	20.26	28.52	**0.065**	30.23	0.304	31.24	1.659	**31.40**	2.498
	0.15	16.97	26.65	**0.068**	28.11	0.292	29.27	1.650	**29.43**	2.499
	0.20	14.76	25.12	**0.059**	26.09	0.300	28.00	2.002	**28.15**	2.884

Table 3: Performance of image denoising for different levels of additive Gaussian noise in terms of averaged PSNR (dB) and averaged execution time (s) on the kodak and Set5 datasets. Best results in bold.

ent features in terms of PSNR for a $\times 2$ upscaling factor (around 0.06dB). Note that this improvements diminishes with higher magnification factors.

The methods included in our testing setup are: bicubic as a baseline, the recently proposed Deep Learning approach of Dong et al. [6] (referred as *SRCNN*), the ANR method of Timofte et al. [20] and its improved version A+ [21] and the novel Super-Resolution forest of [18] (referred as *ASRF*). The datasets and the whole experimental setup is based on the one used in [23, 24, 6, 15], with the addition of the kodak dataset and the $\times 2$ magnification factor.

In order to further test the versatility of our proposed upscaler, we also test it with (clean) depth images, without re-training the regressors with depth content. We use two sequences of 15 images each from the Sintel synthetic depth dataset [2], which we denote as *bambo* and *market.*

Our proposed HHCR has a parallel implementation, and runs in the same CPU as the other methods. We use a K-SVD sparse dictionary of 8192 elements and the chosen cluster size is 4250 k-NN. A+ uses a dictionary of 1024 atoms and a neighborhood size of 2048 atoms, as setting it to 4250 degraded their quality results. In our algorithm the time complexity is affected by the number of hyperspheres used in the SpH configuration rather than the size of the dictionary, therefore we increase its size as our performance scales better than that of A+ both in quality and speed when enlarging the dictionaries. The ASRF forest is trained with the improved alternative training they proposed, with 15 trees and a depth of 16 levels (65536 leaves).

We present two configurations of our algorithm: 1 hypersphere (i.e. exhaustive search) which sets an upper quality limit and 7 hyperspheres which is our optimal configuration in terms of quality vs speed trade-off. By showing both configurations we evaluate the effect of the approximate search both in quality drop and in time speed-up, showing at the same time the full potential of the antipodally invariant search and the IBP features.

We show objective evaluation of all the tested methods in Table 1. First of all, the quality obtained with our HHCR is around 0.2dB higher than that of A+ and ASRF, which often are the second best performers. The speed-up with respect A+ ranges from $\times 4.6$ to 9.3 and about $\times 10$ with respect ASRF. Secondly, the algorithmic speed up by using SpH in the HHC transformed space (i.e. comparison between $s = 1$ and $s = 7$) ranges from $\times 4.8$ to 11 depending on the upscaling factors. The drop in quality is very reduced and ranges from 0.01 to 0.07dB. With $s = 7$ we clearly outperform all the state-of-the-art methods in both running time (with the exception of bicubic) and quality (PSNR). A visual overview of speed vs quality is shown in Fig. 7.

When it comes to depth images, the improvement margins are even broader, as shown in Table 2, where specially for $\times 2$ magnification factor there are improvements up to 0.5dB compared to A+. In Fig. 6 we show some images and zoomed-in crops for subjective evaluation, e.g. the highlighted upper white stripes of the *butterfly* image, where our proposed algorithm reconstruct sharper and thinner edges, obtaining overall better preserved structures.

3.2. Denoising

We also apply our piecewise linear regressor for denoising images corrupted with Gaussian noise of diverse standard deviations σ. As already introduced, we select a first coarse denoising algorithm and let the system learn how to correct the output. We show how the regression stage improves greatly over a very basic estimator, i.e. non-local means (NLM) [1], but also how with a more sophisticated first approximation, i.e. BM3D [4], the regression stages is still able to improve the outcome. We do not aim to obtain state-of-the-art results, but rather demonstrate how our algorithm adapts to different problems with barely any change

		Bicubic		ANR [20]		SRCNN[6]		ASRF [18]		A+ [21]		HHCR, $s = 1$		HHCR, $s = 7$	
	MF	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
Set5	2	33.66	0.002	35.83	0.712	36.34	3.953	36.69	1.264	36.55	0.761	**36.87**	1.167	36.80	**0.105**
	3	30.39	0.002	31.92	0.449	32.39	3.916	32.57	1.050	32.59	0.467	**32.79**	0.583	32.75	**0.080**
	4	28.42	0.002	29.69	0.348	30.09	4.031	30.20	1.059	30.29	0.346	**30.46**	0.382	30.45	**0.075**
Set14	2	30.23	0.002	31.80	1.717	32.18	7.695	32.36	2.114	32.28	1.739	**32.48**	2.223	32.44	**0.205**
	3	27.54	0.002	28.65	0.933	29.00	7.646	29.12	1.670	29.13	0.963	**29.26**	1.075	29.23	**0.153**
	4	26.00	0.002	26.85	0.696	27.20	7.944	27.31	1.371	27.32	0.714	**27.45**	0.716	27.42	**0.155**
Kodak	2	30.85	0.003	32.24	2.938	32.63	13.121	32.76	3.360	32.71	3.161	**32.89**	3.943	32.84	**0.339**
	3	28.43	0.003	29.21	1.615	29.43	12.805	29.63	2.555	29.57	1.678	**29.68**	1.771	29.65	**0.246**
	4	27.23	0.003	27.80	1.199	27.94	13.315	28.17	2.204	28.10	1.226	**28.17**	1.186	28.15	**0.245**

Table 1: Performance of $\times 2$, $\times 3$ and $\times 4$ magnification in terms of averaged PSNR (dB) and averaged execution time (s) on datasets Set5, Set14 and Kodak. Best results in bold.

		Bicubic		ANR [20]		SRCNN[6]		ASRF [18]		A+ [21]		HHCR, $s = 1$		HHCR, $s = 7$	
	MF	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
Bambo	2	31.60	0.003	33.06	3.373	33.84	15.135	34.56	3.608	34.31	3.640	**34.77**	4.259	34.71	**0.382**
	3	29.63	0.003	30.72	1.768	31.26	14.921	31.69	2.623	31.85	1.866	**32.05**	2.226	32.00	**0.278**
	4	28.40	0.003	29.33	1.292	29.75	15.285	30.13	2.221	30.25	1.331	**30.40**	1.392	30.38	**0.267**
Market	2	35.83	0.003	37.46	3.279	38.61	15.053	39.44	3.294	39.04	3.579	**39.59**	4.387	39.53	**0.351**
	3	33.82	0.003	35.11	1.889	35.92	14.643	36.58	2.540	36.73	1.990	**36.95**	1.998	36.91	**0.274**
	4	32.51	0.003	33.62	1.300	34.16	15.661	34.86	2.124	35.00	1.364	**35.21**	1.283	35.18	**0.270**

Table 2: Performance of $\times 2$, $\times 3$ and $\times 4$ magnification in terms of averaged PSNR (dB) and averaged execution time (s) on depth datasets Bambo and Market. Best results in bold.

in its structure.

The features used for the algorithm are straightforward: we use directly the pixel values of patches extracted from the estimator (NLM or BM3D) concatenated with the noisy input patches. We would like to remark that we have not tuned or made further research about the optimality of this feature for the denoising problem, yet the results obtained are promising as to test with features which fit better the denoising problem and benefit from certain synergy with the selected estimator.

The results shown in Table 3 are specially interesting when configured to work with NLM as estimator. The improvement by our regression stage is regularly higher than 1dB, and although the overall PSNR is still lower than more complex approaches as BM3D, it still shows how our method can adapt and respond well for different problems. When applying it to the BM3D, the improvement is upper-bounded around 0.15dB and probably further optimization on the features and characteristics of BM3D output should be taken into account to push the performance forward.

4. Conclusions

In this paper we extend the original framework of Timofte et al. on anchored neighborhood regression [20] and further improve the SpH search stage of Pérez-Pellitero et al. [15]. We analyze the optimality of the metrics involved in piecewise linear regression NN searches, concluding that antipodal invariance has a great impact on the performance of the regression strategy. We introduce the angular metric space as a naturally antipodally invariant solution, but we also present the simple yet effective transformation HHC which preserves antipodal invariance in the Euclidean space. Our proposed transform confines the input points in a half hypersphere obtaining a reliable and inexpensive approximation of a more complex non-linear mapping, as could be MDS. Training regressors and selecting them during testing with antipodal invariant metrics improves neatly the quality performance, and by using our HHC together with SpH, the drop in quality for the approximate search compared to exhaustive search is minimal. We perform several Super-Resolution upscaling tests which widely surpass the most recent state-of-the-art by about 0.2dB in luminance and depth images, while being $\times 9$ faster. We also provide a simple and straightforward configuration to use our proposed framework for denoising images, demonstrating the versatility of the presented work.

Acknowledgment

This work has been partially supported by the project TEC2013-43935-R, financed by the Spanish Ministerio de Economía y Competitividad and the European Regional Development Fund.

References

[1] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, 2005.

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *Proc. European Conf. on Computer Vision*, Part IV, LNCS 7577, pages 611–625, Oct. 2012.

[3] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. In *Proc. European Conf. on Computer Vision*, volume 8693, 2014.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Bm3d image denoising with shape-adaptive principal component analysis. In *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[5] V. de Silva and J. Tenenbaum. Sparse multidimensional scaling using landmark points. *Stanford Technical Report*, 2004.

[6] C. Dong, C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proc. European Conf. on Computer Vision*, 2014.

[7] W. Dong, D. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. on Image Processing*, 20(7), July 2011.

[8] W. Freeman, T. Jones, and E. Pasztor. Example-based super-resolution. *IEEE Trans. Computer Graphics and Applications*, 22(2):56–65, 2002.

[9] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Proc. IEEE Int. Conf. on Computer Vision*, 2009.

[10] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon. Spherical hashing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[11] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2015.

[12] M. Irani and S. Peleg. Improving resolution by image registration. *Graphical Models and Image Processing*, 53(3):231–239, 1991.

[13] H. Li and F. Liu. Image denoising via sparse and redundant representations over learned dictionaries in wavelet domain. In *Int. Conf. on Image and Graphics*, Sept 2009.

[14] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. on Information Theory*, 28(2):129–137, Mar 1982.

[15] E. Pérez-Pellitero, J. Salvador, I. Torres, J. Ruiz-Hidalgo, and B. Rosenhahn. Fast super-resolution via dense local training and inverse regressor search. *Proc. Asian Conf. on Computer Vision*, 2014.

[16] J. C. Platt. Fastmap, metricmap, and landmark mds are all nystrom algorithms. Number MSR-TR-2004-26, page 15, January 2005.

[17] J. Salvador and E. Pérez-Pellitero. Naive Bayes Super-Resolution Forest. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 325–333, 2015.

[18] S. Schulter, C. Leistner, and H. Bischof. Fast and accurate image upscaling with super-resolution forests. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2015.

[19] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2008.

[20] R. Timofte, V. D. Smet, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proc. IEEE Int. Conf. on Computer Vision*, 2013.

[21] R. Timofte, V. D. Smet, and L. V. Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Proc. Asian Conf. on Computer Vision*, 2014.

[22] C.-Y. Yang and M.-H. Yang. Fast direct super-resolution by simple functions. In *Proc. IEEE International Conf. on Computer Vision*, 2013.

[23] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2013.

[24] J. Yang, J. Wright, H. T.S., and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. on Image Processing*, 19(11):2861–2873, 2010.

[25] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Proc. Int. Conf. on Curves and Surfaces*, 2012.