Smooth solution space exploration by introducing local similarity search in a MAP-Elites algorithm

Mauro Ricchiuti[1], Rob Saunders[1] and Alexander Hagg[1]

[1] *LIACS, Leiden University, The Netherlands*

Abstract— MAP-Elites algorithms have been developed and applied with the aim of better illuminating the search space of an evolutionary algorithm by defining an archive of many niches within a behaviour space, where only individuals within a niche compete. Resulting in an archive that illuminates possible solutions, with each niche containing the best performing prototypes found or an area of behaviour space. The space of solutions, however, is not "smooth", i.e., neighbouring niches in the behaviour space of the archive may contain solutions with very different phenotypical traits. The aim of this research was to create an interactive tool to generate and explore archives produced by the MAP-Elites algorithms in a visual context. Smooth archives support interaction by arranging similar phenotypes in proximity to each other, allowing users to more easily navigate them. In order to do so, we have introduced measures of similarity among neighbouring niches and compared the "smoothness" of our models with respect to the standard one. The comparison is presented for two different visual tasks. The results demonstrate that our model, in the proper conditions, keeps the archive diversity and performance of the standard MAP-Elites algorithm, but is able to additionally select the archive prototypes, generating a smoother archive.

Keywords— MAP-Elites, Evolutionary Algorithms, Quality-Diversity, Computational Creativity, Generative Art

I. INTRODUCTION

Creativity and the arts are usually considered unique traits of human beings and are probably one of the main reasons why we consider the living experience to be more than just survival and reproduction, regardless of evidence for any biological purposes and origins [1, 2, 3].

Evolutionary algorithms, inspired by nature, have been used as a tool for artistic exploration and computational creativity research in many different ways, generating interesting results [4, 5] and posing questions and problems [1]. Applications of evolutionary algorithms have mainly focused on performance optimization. Natural evolution, however, can be seen as a process of divergent search which optimizes locally, but at the same time diversifies [6]. A new family of algorithms called quality-diversity (QD), have tried to implement this approach, searching for both quality and diversity at the same time. These types of algorithms aim to create a space of solutions filled with the fittest prototypes for a range of possible behaviours [6]. Early experiments on divergence in evolutionary algorithms, which go under the name of *Novelty Search*, showed that the search for novel behaviours could outperform the search guided by objective fitness [7, 8]. In QD algorithms, the space of solutions is composed of behavioural niches and each niche is populated with the fittest candidate, the prototype, for that specific niche. In this way, the result is an illumination of the fittest among the diverse possible solutions that exist in a certain behaviour space [9]. In the original QD algorithm, novelty search was hybridized with local fitness competition among individuals with similar behaviours, generating a population of local competitions in diverse behavioural niches. This algorithm, called *Novelty Search with Local Competition* (NSLC), developed by Lehman and Stanley [10], showed the intriguing potential of QD algorithms by generating a population of virtual creatures and walking strategies in a single run. Later on, Mouret and Clune [9], developed a new type of QD algorithm called *Multi-dimensional Archive of Phenotypic Elites* (MAP-Elites), which generated a discrete behavioural map to store elite versions of diverse behaviours. This family of algorithms showed very promising results in many fields as robotics and generative adversarial attacks [6]. In 2015, Cully [11] stored in a MAP-Elites archive multiple walking strategies of an hexapod robot, introducing the possibility of querying the right strategy to balance any kind of physical damage. The difference between NSLC and MAP-Elites mainly consists in the way they select the population of solutions and the way the archive is defined. The archive in NSLC is unstructured and keeps track of the best individual, growing the niches over the whole extent of the algorithm. In contrast, the archive in MAP-Elites is fixed and discretized. While in NSLC the most performant individuals (novel and with high local fitness) are selected for the next generation, in the MAP-Elites model the parents are selected randomly from the archive, aiming at filling it as much as possible.

There have been studies which have used QD algorithms, and especially MAP-Elites, in the field of design, videogames and other "creative" domains. MAP-Elites have been used, combined with Deep Neural Network and Compositional Pattern Producing Networks, to generate 3D objects

Contact data: Mauro Ricchiuti, ricchiuti_mauro@libero.it

in which the evolutionary algorithm have been considered as the "creative" component of the framework [12]. Other researches, in which the algorithm was used in an interactive way, have explored the potential of using MAP-Elites as co-creative agents, posing the user in the center of the selection process of the EA and opening the doors to many applications in creative domains [13, 14, 15].

An important component of MAP-Elites is the *Behaviour Characterization* (BC). The BC is a vector which expresses features of an individual's behavior or phenotype [6], it is used to compare novelty among individuals and to locate prototypes in the behavior space, being in this way the diversity driving force of MAP-Elites [6]. The Behaviour Space, however, is a discrete space and individuals from neighbours cells in the archive have no explicit pressure to be similar (both at the phenotype and genotype level) besides the phenotypic traits that define the BC. Indeed, this feature may not have a practical use when the problem is one of optimisation, but it could be an interesting feature if a MAP-Elites algorithm is used as a tool for artistic exploration.

The co-creative model designed by Hagg [16] aimed to interactively evolve an archive of user-selected prototypes. In this work, a modified version of a MAP-Elites algorithm, the Surrogate-Assisted Illumination (SAIL) is used to generate the first batch of prototypes [17]. These prototypes are then fed into a clustering algorithm [18] and the most representative ones in each cluster are then presented to the user for selection. The selected prototypes will then be used as a seed for the next SAIL iteration. In this approach, the aim is to be able to present the prototype in a clean way to the user for their evaluation. In a successive work [14], Hagg and his team introduced a Variational Autoencoder (VAE) in the algorithm. In this model, called *HyperPref*, random genomes are created and evaluated according to a user-defined objective, their phenotypes are then fed to the VAE which reduces the dimensionality and helps determine similarity [14]. Successively, a QD algorithm creates an archive of high-performing solutions and triggers a first intuition of how good solutions should look like. At this stage, the user is able to select the preferred ones, a snapshot of the latent model is taken and the chosen solutions are used to create a new set of solutions by perturbing the original ones and potentially adding new innovations into the data set.

It is important in the context of this research to introduce the notion of *genetic neutrality* [19]. The expression of the genome in an evolutionary algorithm is often a high-dimensional object, e.g., a 2D image, for this reason a low-dimensional representation of the phenotype, the genome, is used as search space. To map the genotype into phenotype, an expression function is used. This function ideally should be bijective, however, often it is not, and different genotypes could be mapped to the same phenotype. We define *genetic neutrality* in the field of evolutionary computation as the phenomena that happens when different genotypes are mapped to the same phenotype [19]. This is one possible source of discontinuity in the MAP-Elites archives, producing neighbour prototypes that are very similar in phenotype but have different genomes. The other one, that is of particular interest in this study, is that very different phenotypes can map to the same area of Behaviour Space, disrupting the smoothness of the archive. In his work [19] Hagg investigated the impact of neutrality on diversity of solutions in the phenotype space, in both cases with or without Domain Knowledge, comparing different types of evolutionary algorithms. He concluded that each approach have strength and weakness and proposed a guide for when to use each approach, depending on whether the goal is to maximize diversity or fitness.

The lack of of an explicit pressure on neighbours to be similar may permit more variation between neighbours than is desirable for interactive applications. Close prototypes could have very different phenotypes, and sometimes successive clustering [16] is needed to present a sorted archive to the user for interactive applications. For some applications of MAP-Elites such as generative design or artistic exploration, could be useful to produce an organized archive, where neighbouring niches share clear similarities beyond those defined by the BC. In these types of applications is important to guarantee the diversity of the archive, keeping in mind that many of these problems might be subjected of genetic neutrality. It may be useful for an artist to generate thousands of different solutions using a QD algorithm and to be able to explore these solutions interactively in real-time. This may be interesting when the solutions are explored in the context of a performance or just as a more user-friendly tool for the artist to choose the prototypes which better fits his needs.

The aim of our research is to evolve an organized MAP-Elites archive, creating a space of solutions which is smooth in the phenotype domain and could be explored interactively for artistic purposes. We approached the problem introducing a measure for local similarity among close niches and studying its impact on fitness and archive diversity. The difference with Hagg's earlier model [16] is that we don't want to interactively evolve the algorithm, but we want it to evolve itself in an organized way, without the need for a successive clustering algorithm, obtaining an easy-to-explore archive. The interactive part in our approach will be used at the end of the MAP-Elites run to smoothly explore the archive. Our objective is not to have control over the prototypes generation as in Hagg's model, but to leave that process to be generative and have a controller to interactively explore the results obtained.

II. METHODS

To explore the impact of genetic neutrality on the generation of MAP-Elites archives, we first conducted experiments without any similarity metrics, defining some "reference" values of archive diversity for our tasks. We used a *Similarity* metric inspired by the *Solow-Polasky Diversity (SPD)* metric [20], and we defined it as the average of the pairwise distances between every niche within an archive, normalized by the maximum possible distance. Equation 2 shows the formula used to compute *Similarity*, while equation 1 illustrate how the Euclidean Distance is computed. We introduced a measure of *Spread* as in Equation 3 defined as the average distance with the closest neighbour normalized by the maximum possible distance. We have also used histograms to inspect the spread of solutions in the space of possibilities. These metrics have been used both to investigate archive diversity and smoothness.

$$d(p,q) = |P-Q| = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{1}$$

where:

p,q = two points in Euclidean n-space
q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)
n = n-space

If not specified, with the word *distance* we will refer to the Euclidean distance between vectors as described in equation 1. Equation 2 and 3 describe respectively the formula used to compute the *Similarity* and *Spread*. In Equation 3, the $q_{closest}$ refers to the closest neighbour to the selected one, as computed by the K-d tree function in `scikit-learn` [21]. The d_{max} value is the maximum possible Euclidean distance in this setting.

$$Similarity = \frac{1}{n \times n}\sum_{i=1}^{n}\sum_{j=i}^{n}\frac{d(p_i, q_j)}{d_{max}} \tag{2}$$

$$Spread = \frac{1}{n}\sum_{i=1}^{n}\frac{d(p_i, q_{closest})}{d_{max}} \tag{3}$$

As in Hagg's work [19], we use Hamming distance as a measure of similarity among close niches to push the system towards producing a smooth archive. This metric has been used as fitness function alone, or combined with other metrics, such as symmetry for a certain class of experiments in which we attempted to produce a smooth archive in optimization problems. Finally, we ran a PCA dimensionality reduction algorithm and we clustered the results using DBSCAN on both genotypes and phenotypes to analyze the diversity of the solution space.

Additionally, a tool has been developed using Derivative TouchDesigner, a commercial node based visual programming language for real time interactive multimedia content, to visually explore the space of solutions in an interactive way.

III. IMPLEMENTATION

We choose to run our experiments on two visual tasks, considering them to yield easy to observe and evaluate results. We developed one task in color space and another one using shapes and symmetry. We have built our models starting from the `pymap_elites` framework [9, 22, 23, 24], a flexible and easy to use MAP-Elites framework for Python, developed by Mouret [25]. Within this library, is possible to specify many different parameters of the EA, like the fitness function, the archive type and the number of niches. Additionally, we have used the CVT version of the algorithm, which uses centroidal Voronoi tessalation to divide the feature space into the desired number of regions and has been proved to be more effective than the standard MAP-Elites for different tasks [22]. We have used `pycairo` for the 2D vector graphics generation, which made it easy to experiment different settings for our tasks. Additionally, we have explored different types of similarity measures, and compare their results.

a. Color Experiments

In the *Hue* experiment, we painted a small canvas 10x10 with a single RGB color using `pycairo`. The genotypes represent red, green and blue colour components. In order to compute the *behavior space* the canvas is re-converted in HSV space by a function from the `PIL` module. The behaviour characterisation of an individual is defined as the *Saturation* and *Value* components of the colour. *Value* and *Saturation* are described in a [0-1] range, while Hue's range is [0-360].

We ran 3 different experiments for 30 times using a different random seed for each run. The first experiment consisted in a standard MAP-Elites run without a fitness function (`f = 0`). This experiment provides a reference for comparison with the following ones. The next 2 experiments introduced a fitness function in terms of similarity among neighbouring niches computed as the hue distance between the 2 canvas colours. This value was measured as descibed in equation 4 considering the hue of the current prototype and $\mathrm{hue}_{\mathrm{nei}}$ the one of the selected neighbour. In the *MaxDist* version the fitness was computed as the maximum distance among niches as shown in equaion 5 while in the other experiment it was computed as the average among the distances of close niches as in equation 6 and we will refer to this experiment as *Average*.

$$Hue = \frac{min(abs(hue - hue_{nei}), 360 - abs(hue - hue_{nei}))}{180} \tag{4}$$

$$Fitness_{MaxDist} = argmax(Hues) \tag{5}$$

$$Fitness_{Avg} = mean(Hues) \tag{6}$$

We experimented with different parameters for this task to explore the effect of the number of neighbour niches used to calculate the similarity fitness, the effect of the number of initialized individuals and the role of the ancestors. Section IV provides an in-depth discussion on the metrics and results.

b. Shape Experiments

The shape experiment setup took inspiration from a similar visual task designed by Hagg et al [26]. Here, we draw on a 100x100 canvas, using 8 points in polar coordinates. Each point has an initial angular position, which distributes them around the center, such that, each new point is rotated 45 degrees from the previous one, and have a radius in the range [0.1-1]. Each point is defined by 2 values, the length of the radius, and the angle formed with its initial point position. The 16 values defining the 8 points is the genome of our EA.

The *behaviour characterisation* is defined as the width and length of the generated shape, and is computed by a blob tracking algorithm in `openCV`. We choose to use the Hamming distance from the `scipy` module, as our measure of *Similarity*, which measures the number of positions in which the pixels of the rendered shapes are different a sequence [27].

Definition (Hamming Distance) 1 *Given two binary vectors $u, v \in Fn$ we define the hamming distance between u and v, d(u, v), to be the number of places where u and v differ.*

We did multiple runs of this experiment, with different setups. We ran a *standard* experiment, with no fitness function, to generate a reference model and multiple experiments changing the way the distance is computed (*average* or *maximum* among close niches). We finally introduced a fitness function as the degree of Symmetry of the shape computed as the sum of the differences between each pair of opposite points. Equation 7 shows the formula used to compute *Symmetry*. In this formula r_i and θ_i represent respectively the radius and the angle of the points in polar coordinates while r_{iop} and θ_{iop} are their opposite ones.

$$\begin{aligned} Symmetry = -\sum_{i=0}^{4} &|r_i sin(4\pi\theta_i) - r_{iop} sin(4\pi\theta_{iop})| \\ &+ |r_i cos(4\pi\theta_i) - r_{iop} cos(4\pi\theta_{iop}))| \end{aligned} \tag{7}$$

In the Symmetry experiments, Similarity and Symmetry are summed linearly to combine them and use them as fitness function.

IV. RESULTS

a. Hue experiment

Figure 1 shows a plot of the CVT archive produced by a run of the standard MAP-Elites algorithm without any fitness function or similarity measure. The vertical axis is the *Value* component of the Behavior descriptor and the horizontal axis is the *Saturation* value of it. Each niche is displayed with the fittest phenotype, i.e., colour, found for that niche. This plot illustrates the reference used to compare results against the constrained *MaxDist* and *Average* versions.

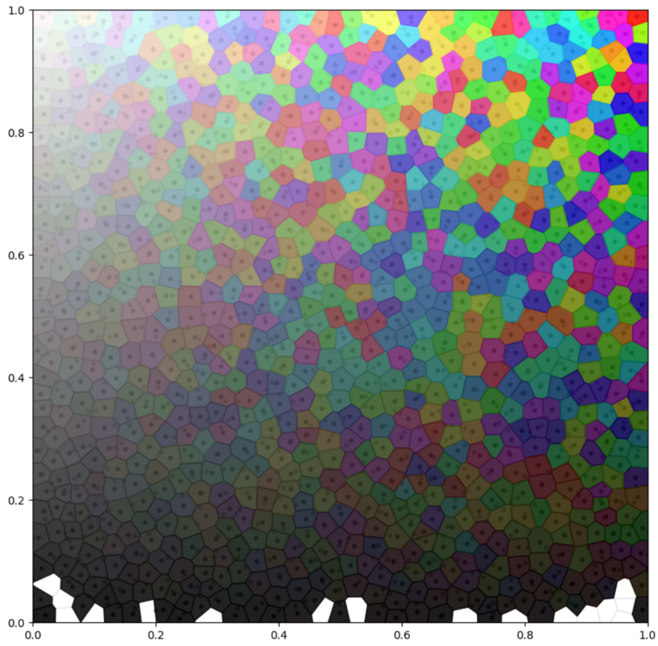
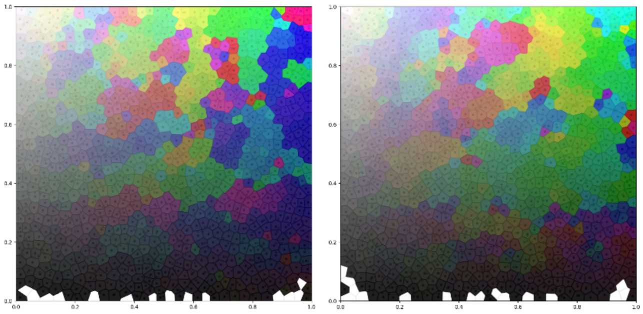
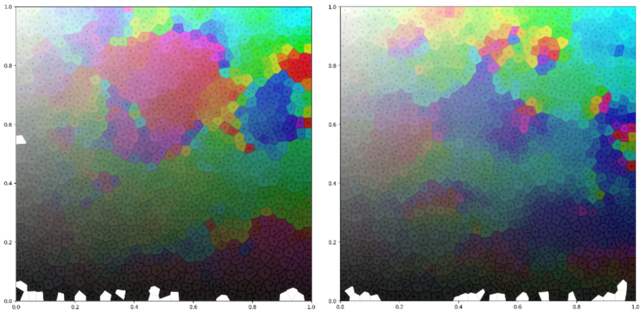


Fig. 1: HSV plot of the standard MAP-Elites algorithm without any measure of similarity or fitness function

Figure 2 shows the results of the *MaxDist* experiments, computing the similarity measure among 3, 4, 5 and 6 close neighbours. We observed that the regions of similar colours increased with the number of neighbours compared. Based on this observation, and to balance the goal of producing regions of smoothly changing hues with the computational cost of computing similarities, we decided that 7 neighbours was a good number.

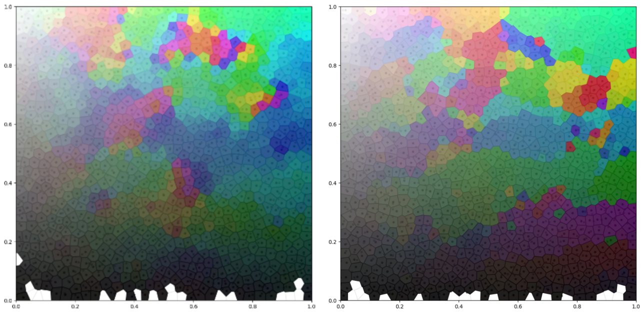


(a) MaxDist3 (b) MaxDist4



(c) MaxDist5 (d) MaxDist6

Fig. 2: Hue experiment computing the similarity as maximum distance among 3,4,5 and 6 neighbours



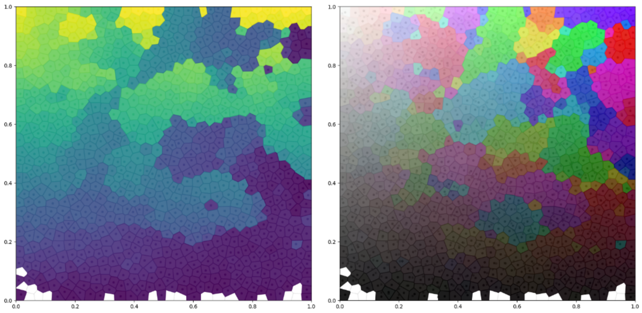
(a) MaxDist7 (b) Avg7

Fig. 3: Hue experiment computing the similarity as maximum distance (a) and average distance (b) among 7 close niches

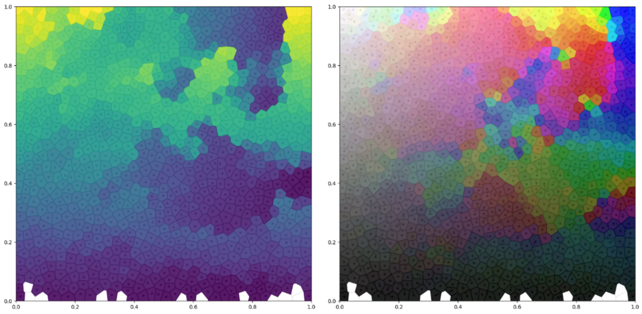
Using 7 neighbours to compute similarity, we also performed the *Average* experiment as described in the previous section. The results of the experiments comparing the *MaxDist* and *Average* experiments using 7 neighbouring niches are shown in Figure 3. It is noticeable that the *Average* model exhibits more distinct edges between clusters of niches and that the clusters of similarly coloured niches vary less, producing the visual impression of flat areas of very similar hues with sharp edges between them. This tendency of the *Average* algorithm to “flatten” the Hue value in a cluster might be explained by the fact that it is taking into account all the 7 close niches when computing the similarity, the *MaxDist* version, on the contrary, is taking into account just the biggest distance among the 7 closest ones.

This observation motivated further analysis to try to better understand the process that generated this result. In order to do so, we recorded the number of ancestors for the prototype (winning) individual in each niche. We did this by generating a counter for every prototype, assigning the maximum counter value among the 2 parents of the new offspring, and increasing that counter every time the prototype is added to the archive. In this way, at the end of the run, each niche stores a value representing the cumulative generation's number. Results of this experiments are shown in Figure 4. For this visualization, each niche is rendered with a color corresponding to its number of ancestors. Here, the blue to yellow gradient represent respectively regions from low to high numbers of ancestors.

Some regions clearly exhibits correlations between number of ancestors and HSV colors. Clusters of same number of ancestors are expressed as zones with similar Hues and the boundaries of these clusters are sometimes visible both in the ancestors plot and in the HSV color plot. This is even more visible in the *Average* experiment because its HSV plot already shows sharp edges. Regions in which the boundaries do not correspond in the 2 figures, might be considered cases of spreading of hue in neighbour niches.



(a) Ancestors Avg7 (b) Avg7



(c) Ancestors MaxDist7 (d) MaxDist7

Fig. 4: Ancestors experiment (a) and (c) illustrate the number of ancestors of each niche, being blue the regions of low number of ancestors and yellow region with high number of ancestors, and (b) and (d) illustrate the respective HSV archives

In order to have a statistical metric of child-parent relationship in the Hue experiment, we computed the average child-parent distance in Behavior Space of the last generation of prototypes in the archive. We did this by storing the parents pair of every niche in the archive and then computing the Euclidean distance between their Behavior Descriptor and the one of the respective prototype. The results are shown in Table 1. We ran the ANOVA test on both parents distances obtaining f-values of 127 and 138 and p-values of 1.3×10^{-28} and 7.9×10^{-31} for the MaxDist experiment, f-values of 181 and 211 and p-values of 1.6×10^{-39} and 1.7×10^{-45} for the Avg experiment rejecting the null hypothesis. Table 1 clearly illustrate that our approaches have shorter child-parent distances in behavior space. This means that new offspring are usually situated in Behavior Space closer to their parent, compared with the *Standard* version in which they are more spreaded in the solution space. This results did not surprised us, and we attributed them to the local similarity pressure we introduced.

	Standard	MaxDist7	Average7
Parent 1	0.43344	0.23734	0.26137
Parent 2	0.43193	0.23464	0.26430

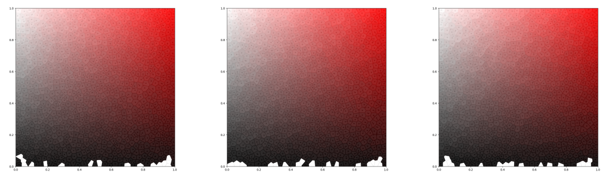
TABLE 1: Child-Parent average distance

We assessed the *Diversity* of the archive by computing measures of *Similarity* and *Spread* as introduced in the Methods section. We ran the experiments 10 times for 100K evaluations each changing the random seed value at each run to avoid bias from the random distribution. We evaluated *Similarity* and *Spread* both at the phenotype and genotype level. The results are shown in Table 2. We can conclude that our modification to the algorithm isn't affecting the *Similarity* metric, so we can assume that the archive is keeping its Diversity. The *Spread* however, is dramatically reduced as expected by introducing a local measure of similarity. It is interesting to notice that even if our measure of similarity as fitness function is applied on the phenotype level, both genotype and phenotype's *Spread* are significantly affected by it. This task differs from the Shape experiment in the neutrality domains. The Hue experiment exhibits genotype-behavior neutrality since a prototype having same Value and Saturation's values but different Hue, would be assigned to the same archive location. To better illustrate this neutrality, we plotted in Figure 5 the archives for the 3 different experiments setting the Hue to a constant value of 1. In this way it is visible that the 3 archives are really similar when considering only the behavior space. On the contrary, the Shape experiment shows a genotype-phenotype neutrality.

	Standard	MaxDist7	Average7
Similarity Geno	0.34234	0.33702	0.3368
Similarity Pheno	0.37956	0.37169	0.37199
Spread Geno	0.13241	0.03998	0.03118
Spread Pheno	0.16653	0.06297	0.05028

TABLE 2: Average and Normalized values of Similarity and Spread computed on the Genotype and Phenotype vectors

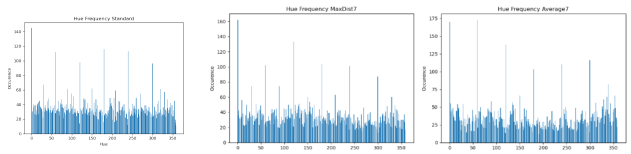
The diversity of the archive has also been studied by comparing the Hue distributions of the 3 different experiments. Hue values are included in the range [0-360] and plotted in Figure 6 . Finally we computed the average filling percentage of the archives. The results showed that all the 3 experiments setups managed to almost completely fill the archives, being 98.31%, 98.29% and 98.49% for respectively the *Standard*, *MaxDist7* and *Average7* experiments.



(a) Standard (b) MaxDist7 (c) Average7

Fig. 5: Plot of the archive setting the Hue to a constant value of 1 (a) Standard, (b) MaxDist7 and (c) Average7 setups

We can affirm that for this task, the introduction of a similarity measure among neighbour niches did not impact the Diversity of the archives but positively reduced the *Spread*, generating a smoother archive.



(a) Standard (b) MaxDist7 (c) Average7

Fig. 6: Distributions of all Hue's values of the 10 experiments for (a) Standard, (b) MaxDist7 and (c) Average7 setups

b. Shape Experiments

The shape experiment has been implemented as described in the previous section. We ran this experiment to demonstrate that the approach generalises beyond the Hue experiment, where the neutrality is due to the mapping from genotype to behaviour, due to the removal of hue in the behaviour space, whereas in this experiment the neutrality is due to the mapping from genotype to phenotype, which attempts to reflect more real-world applications. Additionally, in this experiment we introduced Symmetry as fitness measure, which was not present in the previous task. Here, instead of computing the *Similarity* and *Spread* as in the previous experiment, we computed the Hamming distance between each prototype. Table 3 reports the *Similarity* and *Spread* for the experiments using only Similarity (No Symmetry) as fitness function and the ones using both Similarity and Symmetry (Symmetry).

	Symmetry		No Symmetry	
	Similarity	Spread	Similarity	Spread
Standard	0.1316	0.0772	0.1844	0.1265
MaxDist3	0.078	0.0487	0.1370	0.0601
MaxDist7	0.085	0.0507	0.1111	0.0404
AvgDist3	0.1145	0.0654	0.1574	0.0821
AvgDist7	0.1151	0.0574	0.1924	0.1209

TABLE 3: Similarity and Spread computed using Hamming distances between prototypes, both for the experiments using only Similarity as fitness function (No Symmetry) and the one using both Similarity and Symmetry(Symmetry)

The measures shown in Table 3 were computed using the *Hamming Distance* metric provided by the `scipy.spatial.distance.hamming` function, feeding it with a flattened version of `numpy` array holding the pixel's values. Since we also used the *Hamming Distance* as a measure of Similarity in our algorithm, to confirm our results, we ran the same analysis computing the *Euclidean* distances instead of the *Hamming* distances. The results are shown in Table 4 and show the same correlation as shown in Table 3.

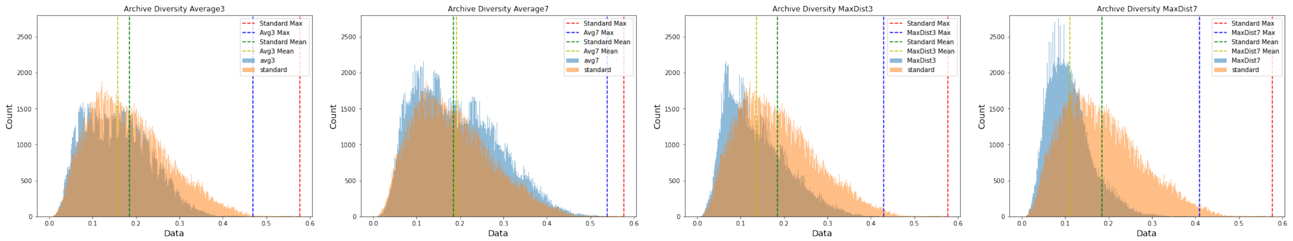
Both the *Average* and *MaxDist* experiments show smaller Similarity and Spread values for all the computed metrics compared with the *Standard* approach. The *MaxDist7* version exhibits the smallest *Spread* and *Similarity* distances for No Symmetry experiment, and we can interpret these results as a decrease of *Diversity* in the archive. The *Average7* experiment preserves the distance between prototypes yielding *Spread* and *Similarity* values close to the one computed on the *Standard* model. In the No Symmetry experiment, *Average7* produced even a higher value of Similarity in respect of the Standard version. For the experiments using a combination of Similarity and Symmetry as fitness function, the *Average3* algorithm seems to manifest the best performances in terms of keeping the distance between prototypes and so the *Diversity*, showing the highest Spread value and a considerably high Similarity value.

	Symmetry		No Symmetry	
	Similarity	Spread	Similarity	Spread
Standard	0.3096	0.1979	0.3811	0.2928
MaxDist3	0.2170	0.1362	0.3170	0.1706
MaxDist7	0.2286	0.1352	0.2796	0.1150
AvgDist3	0.2783	0.1734	0.3471	0.2129
AvgDist7	0.2827	0.1552	0.3877	0.2854

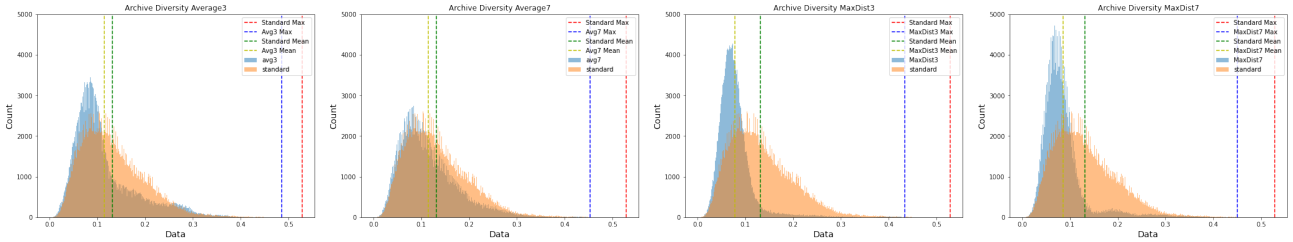
TABLE 4: Similarity and Spread computed using Euclidean distances between prototypes, both for the experiments using only Similarity as fitness function (No Symmetry) and the one using both Similarity and Symmetry(Symmetry)

We plotted the distances distribution in the form of histograms to visually evaluate the *Diversity* of the archives. The results are shown in Figure 7, in which the orange plots represents the distribution of the distances for the *Standard* model, and the blue for our versions of the algorithm. Figure 7(b) illustrates that the *Average7* model seems to be the one preserving the *Diversity* of the archive, and this was also shown in the results in Table 3 and 4, exhibiting the maximum values of *Similarity* among our versions. This is also the case for the version introducing Symmetry as fitness function, and Figure 7(f) shows two almost overlapping histograms. The *MaxDist7* again is confirmed to be the worst performing in *Diversity* for the Similarity only experiment as shown in Figure 7(d), while the *MaxDist3* appears to have the lowest diversity in the Symmetry experiment.

It is interesting to notice in Figure 7 that the distribution of our algorithm is not a simple normal distribution, but exhibits a small peak around 0.28 for Figure 7(e) and 0.2 for Figure 7(h). Additionally, some elements could still be seen towards the right sides of the plots in almost all the version. This could be interpreted as a good sign of the algorithm still preserving some *Diversity*. The prototypes seems to be concentrated in some regions due to the similarity pressure, however, the algorithm is still able to discover farther prototypes which covers almost the whole distance spectrum as in the *Standard* version. Figure 7 shows also the mean and maximum values of the distribution. It is interesting to notice that



(a) Avg3 Sim (b) Avg7 Sim (c) MaxDist3 Sim (d) MaxDist7 Sim



(e) Avg3 Sim+Sym (f) Avg7 Sim+Sym (g) MaxDist3 Sim+Sym (h) MaxDist7 Sim+Sym

Fig. 7: Hamming Distances Histograms for the experiments using only Similarity (a, b, c, d), and the combination of Similarity and Symmetry (e, f, g, h)

the maximum distance in the archive for our version is lower then in the Standard model, but it is far away from the mean value. In Figure 7(g) for example, even if the distribution is compacted in a narrow shape, the maximum distance computed is far outside the main shape. This could be interpreted as the algorithm keeping the *Diversity* of the archive.

Additionally, we developed in Derivative TouchDesigner a tool to explore the archive to both inform intuition and as a proof-of-concept for an interactive tool to use with such archives. The output of the tool is visible in Figure 8 and shows the plots for the 5 different versions of the algorithm. For each version, a prototype is selected and plotted in red, using the 2D slider on the bottom right of figure 8 which directly maps the behavior space. The 7 closest neighbouring niches are queried from the k-D tree and plotted in black in relative behavior space position.

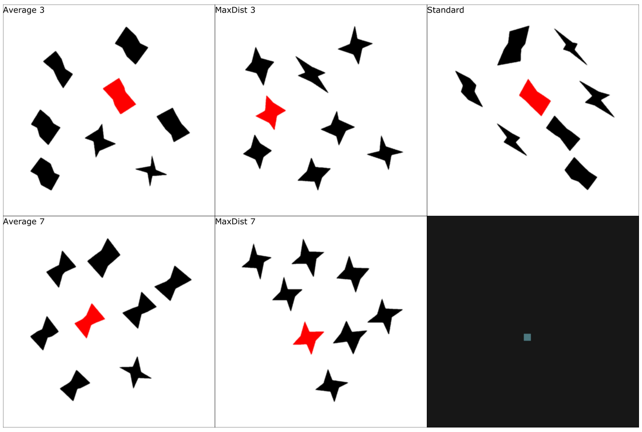
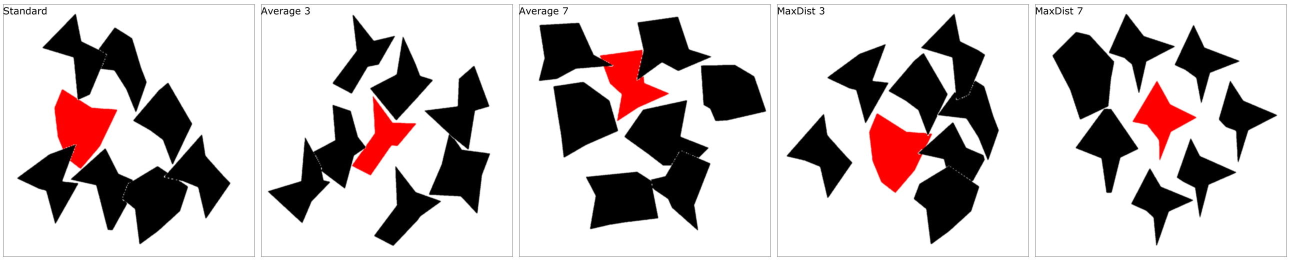


Fig. 8: Plots of the different versions of the algorithm as appear in the TouchDesigner implementation. On the bottom right side, the 2D slider used to explore the archive is visible

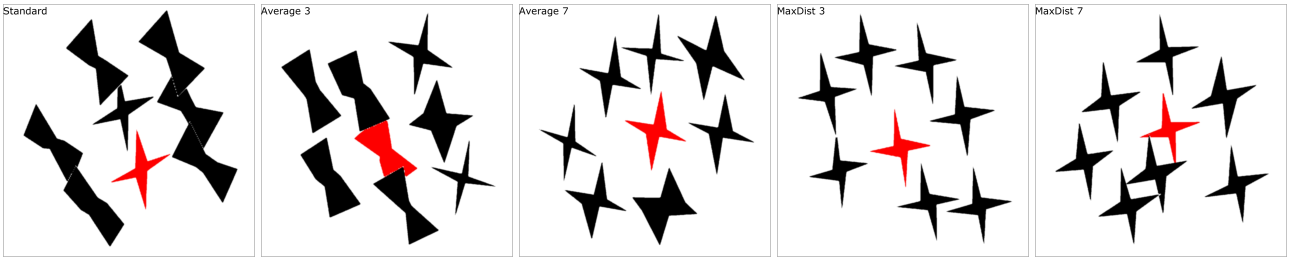
Figure 9 shows random sampled part of the archive for the Similarity and Similarity + Symmetry versions. The red shape is the selected prototype, while the other 7 shapes are the closest neighbouring prototypes in the archive as computed by a k-d tree search. They are placed in their relative normalized behavior space position in respect to the selected prototype in order to show their spatial relation with the selected instance. These figures are screenshots from the interactive tool we developed in TouchDesigner, and are exemplary of the output of it. We explored the archive using a 2D slider, in which the (x,y) coordinates were used as position in Behavior space to query the k-d search in real-time. Few general conclusions can be drawn from Figure 9, because each sample provides a small, local snapshot of the archive. However, they appear to support the earlier observation from the colour experiment that the *MaxDist3* and especially the *MaxDist7* experiment with Sim+Sym fitness function yield similar prototypes among close neighbours. For all the settings, the Similarity only version seems to yield more variety as a consequence of the lack of the Symmetry component in the fitness function. However, we noticed that even in this case, the *MaxDist7* reduces the variety among close niches. The *Average7* experiment seems to generate quite similar prototypes on a local level, but moving through the archive, it shows a good level of variety. These observations match with the results in Table 3, 4 and Figure 7 which indicated that *Average7* maintained a similar level of variety compared to the *Standard* model. Additionally, our observation showed that the *Average7* algorithm was capable of producing very similar prototypes on a local level, generating a smoother archive compared with the other approaches in the No Symmetry experiment.

To better study the clustering behavior of the different experiments, we ran a Principal component analysis (PCA) dimensionality reduction algorithm, followed by a DBSCAN clustering algorithm using different settings. We ran it both on phenotype and genotype looking for differences due to the *genetic neutrality*, but we didn't saw intresting results. We also ran the algorithm using Euclidean distance metric to avoid bias, being the Hamming distance already been used as the Similarity metric.

We noticed that the number of clusters found on the geno-



(a) Standard (b) Avg3 (c) Avg7 (d) MaxDist3 (e) MaxDist7



(f) Standard(Symmetry) (g) Avg3(Symmetry) (h) Avg7(Symmetry) (i) MaxDist3(Symmetry) (j) MaxDist7(Symmetry)

Fig. 9: Shape experiments plot, rendering close neighbours in relative behavior space position to the selected prototype(red) for the "No Symmetry" experiment (a-e) and the "Symmetry" one (f-j)

type is bigger than for the phenotype for every of our experiments, but this is not valid for the Standard experiments for both the Similarity only and Similarity+Symmetry experiments, in which the number of clusters in the phenotype is bigger than in the genotype.

Figure 10 shows the plots of the DBSCAN clustering algorithm on the results of the PCA applied to the phenotype of the experiments using only Similarity and Similarity+Symmetry. There are no significant differences in the 2 experiments since the number of clusters found are very similar. It may be interest to notice the difference in the number of no-clustered (outliers) samples between the Standard experiments and the other ones. These values are summarized in Table 5 and they show that in the case of the Standard model, for both experiments, the number of outliers is greater than for the other experiments. The values used to determine clusters in the DBSCAN algorithm vary for every experiment, so it is not guaranteed that the comparison between these number is valid. However, they might be interpreted as a good indicator that our approach is effectively working in generating more similar prototypes.

	Similarity	Similarity+Symmetry
Standard	193	109
Average3	56	47
Average7	104	38
MaxDist3	85	80
MaxDist7	83	76

TABLE 5: Number of outliers (not clustered samples) for the Similarity and Similarity+Symmetry experiments

V. CONCLUSIONS

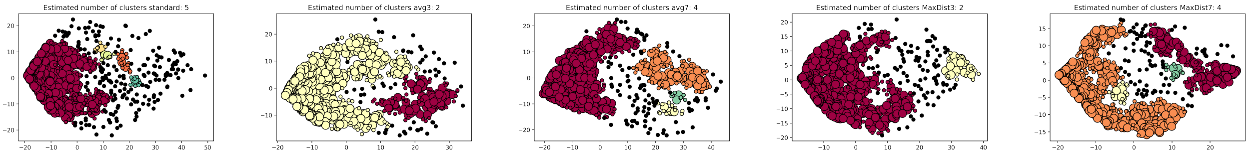
With some of our setups, we did manage to obtain smoother but equally diverse archives compared to the standard MAP-Elites algorithm, showing that this technique could improve the prototypes arrangement in the archive in the generation phase without needing further sorting or clustering post-processes. Our algorithm still does not manage to achieve a level of similarity among close niches such that the exploration of the archive with our interactive tool is totally smooth. However, we can say that this research could be a starting points towards the development of future researches on smooth MAP-Elites archives.

MAP-Elites may not be the best approach for generating interactive artistic tools for solution exploration, and the final result of this research is still far from being a complete and usable tool. However, we think that the use of MAP-Elites as an artistic tool may be promising in the future, and could be useful for example in real-time contexts in which the artist could explore a space of different solutions in different domains other than the visual one.

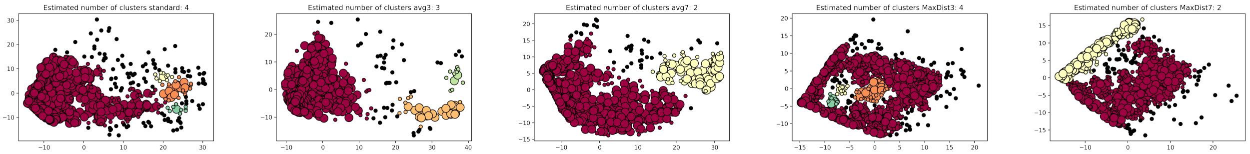
We think that our approach could be beneficial in problems which present lots of neutrality and in which having similar solutions located in the archive could be utilized as variations in behavior (i.e. problems of locomotion in robots). Could be interesting to test the algorithm on such problems, in order to explore its effects on other domains besides the visual one. Further works could try to improve this feature for example by applying different similarity measures or using dimensionality reduction algorithms as a pre-process step to determine similarity, making it more effective. This research shows a new way of working with MAP-Elites and tries to give a new approach to tackle problems involving genetic neutrality. Additionally it stimulate the discussion on the possibilites of using MAP-Elites as interactive tools opening up many possibilities of future applications of Quality-Diversity algorithms in creative domains.

REFERENCES

[1] J. McCormack, "Open problems in evolutionary music and art," in *Workshops on Applications of Evolutionary Computation*. Springer, 2005, pp. 428–436.



(a) Standard Sim (b) Avg3 Sim (c) Avg7 Sim (d) MaxDist3 Sim (e) MaxDist7 Sim



(f) Standard Sim+Sym (g) Avg3 Sim+Sym (h) Avg7 Sim+Sym (i) MaxDist3 Sim+Sym (j) MaxDist7 Sim+Sym

Fig. 10: DBSCAN clustering algorithm applied on PCA results of phenotype for both the versions using only Similarity(Sim) and Similarity+Symmetry(Sim+Sym)

[2] E. Dissanayake, *What is art for?* University of Washington Press, 2015.

[3] G. Miller, *The mating mind: How sexual choice shaped the evolution of human nature*. Anchor, 2011.

[4] J. Secretan, N. Beato, D. B. D Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley, "Picbreeder: evolving pictures collaboratively online," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1759–1768.

[5] K. Sims, "Artificial evolution for computer graphics," in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, 1991, pp. 319–328.

[6] J. K. Pugh, L. B. Soros, and K. O. Stanley, "Quality diversity: A new frontier for evolutionary computation," *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016.

[7] J. Lehman, K. O. Stanley *et al.*, "Exploiting open-endedness to solve problems through the search for novelty." in *ALIFE*. Citeseer, 2008, pp. 329–336.

[8] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary computation*, vol. 19, no. 2, pp. 189–223, 2011.

[9] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.

[10] J. Lehman and K. O. Stanley, "Evolving a diversity of virtual creatures through novelty search and local competition," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 211–218.

[11] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.

[12] J. Lehman, S. Risi, and J. Clune, "Creative generation of 3d objects with deep learning and innovation engines," in *Proceedings of the 7th International Conference on Computational Creativity*. Citeseer, 2016, pp. 180–187.

[13] A. Hagg, A. Asteroth, and T. Bäck, "Modeling user selection in quality diversity," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 116–124.

[14] ——, "A deep dive into exploring the preference hypervolume." in *ICCC*, 2020, pp. 394–397.

[15] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, "Empowering quality diversity in dungeon design with interactive constrained map-elites," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019, pp. 1–8.

[16] A. Hagg, A. Asteroth, and T. Bäck, "Prototype discovery using quality-diversity," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2018, pp. 500–511.

[17] A. Gaier, A. Asteroth, and J.-B. Mouret, "Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 99–106.

[18] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[19] A. Hagg, M. Preuss, A. Asteroth, and T. Bäck, "An analysis of phenotypic diversity in multi-solution optimization," in *International Conference on Bioinspired Methods and Their Applications*. Springer, 2020, pp. 43–55.

[20] A. R. Solow and S. Polasky, "Measuring biological diversity," *Environmental and Ecological Statistics*, vol. 1, no. 2, pp. 95–103, 1994.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] V. Vassiliades, K. Chatzilygeroudis, and J.-B. Mouret, "Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 623–630, 2017.

[23] V. Vassiliades and J.-B. Mouret, "Discovering the elite hypervolume by leveraging interspecies correlation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 149–156.

[24] J.-B. Mouret and G. Maguire, "Quality diversity for multi-task optimization," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 2020, pp. 121–129.

[25] J. B. Mouret, "pymap_elites," https://github.com/resibots/pymap_elites, 2020.

[26] A. Hagg, S. Berns, A. Asteroth, S. Colton, and T. Bäck, "Expressivity of parameterized and data-driven representations in quality diversity search," *arXiv preprint arXiv:2105.04247*, 2021.

[27] B. Waggener, W. N. Waggener, and W. M. Waggener, *Pulse code modulation techniques*. Springer Science & Business Media, 1995.