Formally Testing Liveness by Means of Compression Rates*

César Andrés, Ismael Rodríguez, and Fernando Rubio

Dept. Sistemas Informáticos y Computación
Facultad de Informática, Universidad Complutense
c.andres@fdi.ucm.es,{isrodrig,fernando}@sip.ucm.es

Abstract. We present a formal method to determine whether there exist *living* creatures in a given computational environment. Our proposal is based on studying the evolution of the entropy of the studied system. In particular, we check whether there exist entities decreasing the entropy in some parts, while increasing it in the rest of the world, which fits into the well-known *maximum entropy production principle*. The entropy of a computational environment is measured in terms of its *compression rate* with respect to some compression strategy. Some life-related notions such as *biodiversity* are quantified as well. These ideas are presented by means of formal definitions. A toy example where a simple living structure is identified in a video stream is presented, and some results are reported.

Keywords: Artificial Life, Maximum Entropy Principle, Compression Algorithms.

1 Introduction

Whenever the important question of *what is life* is considered, the controversy eventually arises. For elementary school students, the answer is rather simple: Live beings are those which feed themselves, relate with the environment, and reproduce. However, this definition is neither operative nor precise enough in practice. On the one hand, defining notions such as *feeding*, *relating*, and *reproducing* with enough generality to embrace all kinds of living beings existing in Nature is not easy. Moreover, if Artificial Life is considered [1,5,7,8,16], then defining these concepts is even more challenging. On the other hand, the previous definition of life ignores some living beings that do not fulfill some of the proposed conditions (e.g., *mules* do not *reproduce*).

In this regard, we may consider the Maximum Entropy Production Principle (see e.g. [2,3,9,11]). Grossly speaking, this principle states the following ideas: (a) Due to the Thermodynamics laws, the entropy of any environment must increase along time; (b) living beings are repetitive patterns that increase the order in their environment by their simple existence: Species are made of repetitive patterns (living individuals), and the parts of a living being are repetitive themselves (organs, cells, etc); so, (c) if (a) and (b) are not contradictory then living beings must generate *more* entropy around them than the entropy reduced by the existence of their bodies themselves. That is, living beings are entities with low entropy that increase the entropy around them as they live.

* Supported by projects TIN2006-15578-C02-01, PAC06-0008-6995, MRTN-CT-2003-505121.

G. Rudolph et al. (Eds.): PPSN X, LNCS 5199, pp. 347–357, 2008.

Let us note that the definition of entropy actually depends on the kind of environment we are considering. In a chemical environment, Thermodynamics provides appropriate entropy notions. In an information environment, several notions of *order* and *entropy* are available. For instance, Shannon's Theorem [12] provides a classical definition of entropy. This notion is restricted to a *memoryless* source, i.e. the probability of each symbol is assumed to be independent of the symbols entering the remaining sites in the chain. While being appropriate for some cases, this assumption is unrealistic in most approaches. More generally, given two sequences of bits α_1 and α_2, we can find two formal criteria C_1 and C_2 to measure the information entropy such that α_1 is more *ordered* than α_2 for C_1, and it is the other way around for C_2. For instance, let us suppose that C_1 (respectively, C_2) measures the order degree of a sequence of bits in terms of the *compression rate* we achieve by applying a compression algorithm A_1 (resp. A_2) to the sequence. If the resulting compressed sequence is *short* then it means that the compression algorithm finds repetitive patterns and regularities in the original sequence, i.e. the original sequence is highly ordered. If the compressed sequence is long then the original sequence represents a chaotic piece of information, i.e., a high entropy environment is detected. Depending on whether A_1 or A_2 are applied, some kinds of patterns will be detected as repetitive while some others will not (actually, there does not exist any perfect compression strategy). Hence, if a general approach is considered then the information entropy is a relative notion indeed.

In this paper we present a formal framework to detect living beings in information streams. Following the Maximum Entropy Production Principle, we seek for low entropy structures that increase the entropy around them. Compression algorithms are used to define *order* and *chaos* in each case. In fact, the proposed method to detect life is parameterized by the definition of entropy we wish to consider, i.e. by the specific *compression algorithm* we are considering. Several formal notions to detect life and classify it, as well as to assess the *biodiversity* of the analyzed environment, are considered. In addition, a toy example is considered and some experimental results are reported. In particular, we search for structures fulfilling our definition of life within a video stream representing an execution of the classical *Snake* game.

The rest of the paper is structured as follows. In the next section we present some preliminary concepts and we use them to define Life in terms of compression rates according to the Maximum Entropy Production Principle. Besides, we present some concepts concerning the biodiversity of artificial ecosystems, and we deal with the notions of births and deaths in our framework. In Section 3, we present an example where we apply some of the proposed concepts to detect life in a game execution. Finally, some conclusions and future work are given in Section 4.

2 Formal Model

In this section we present some basic notions to define and manipulate information in our framework. We denote by *world* the information source where we search for living structures. In formal terms, a world is a set of points located in the space and time where each point has attached a binary value. We represent these sets by means of a function, as shown below.

Definition 1. An $n-$*world* is a partial function $w : \mathbf{N}^n - \rightarrow \{0, 1\}$ with finite domain. We say that the *scope* of w is the domain of w, and we denote it by S_w.

We denote by `n-Worlds` the set of all possible $n-$worlds. □

Let us note that the previous definition uses n dimensions without making any special difference to represent the time. In fact, we may assume that the time is just one of the n dimensions. Thus, we can trivially represent *dynamic* worlds evolving in time. Next we define the parts of a world. This notion will be required later to identify living structures within a world. In addition, we represent some algebraic operators that will be used to combine worlds.

Definition 2. Let w, w' be two $n-$worlds. We say that w' is a *subworld* of w, denoted by $w' \subseteq w$, if $S_{w'} \subseteq S_w$ and for all $x \in S_{w'}$ we have $w'(x) = w(x)$.

Let w_1, w_2 be two $n-$worlds such that $S_{w_1} \cap S_{w_2} = \emptyset$. The *union* of worlds w_1 and w_2, denoted by $w_1 \cup w_2$, is a new world w with scope $S_w = S_{w_1} \cup S_{w_2}$ such that $w(x) = w_1(x)$ if $x \in S_{w_1}$ and $w(x) = w_2(x)$ if $x \in S_{w_2}$. □

Once we can deal with subparts of a world, we can present some preliminary notions to identify repetitive patterns inside it. Since living beings are parts of the world where they exist, we can use subworlds to delimit those parts of the world that actually denote living structures.

Let us note that if a given structure appears several times then we can *codify* the presence of all its instances with a representation shorter than if these structures were different. Thus, repetitive patterns allow to reduce the length of the codification of the whole world where they are. If we consider this argument the other way around, repetitive patterns can be identified in a world by applying a compression algorithm to the world. Essentially, a compression strategy is just a codification. That is, it is a transformation of a world into a sequence of bits. These transformations induce a *compression rate*, that is, the rate between the length of the compressed sequence of bits and the size of the world.

Definition 3. A *compression strategy* for $n-$worlds is a function C where we have $C : \texttt{n-Worlds} \rightarrow \{0, 1\}^*$.

Let $w \in \texttt{n-Worlds}$. The *compression rate* of C for w is defined as $\frac{\texttt{length}(C(w))}{\mid S_w \mid}$, and it is denoted by $\texttt{CompRate}(C, w)$. □

Let us note that we are considering a very general notion of compression strategy. For instance, if we restricted ourselves to e.g. Huffman codes [4] or algorithms such as LZW [14] then the generality of the framework would be reduced. In contrast, searching for redundancies in several different ways is allowed in the proposed framework. For instance, we may search for decompositions of frequencies by using the discrete Fourier transformation or the discrete cosine transformation (see e.g. [6,10]) (in particular, the JPEG transformation will be considered in the example presented at the end of the paper). The generality of the previous definition will allow us to search for life patterns in a broader sense than usual. In particular, the criterion to detect repetitive patterns will depend on the particular compression strategy considered in each situation. Thus, a pattern under a certain strategy could not be a pattern under another one. This reflects the fact that the interpretation given to the world strongly depends on the *rational model* we use to describe that world. In our case, the considered rational model is denoted by the compression strategy.

Next we define what a (living or not) repetitive pattern is. A *pattern* is a subworld such that, when it is considered as part of a given world, the compression rate of the world is reduced. In other words, a pattern allows to increment the order of the world where it is inside. A subworld can be a pattern due to two different reasons. On the one hand, a subworld can help to form redundancies in the world it belongs to. In this case, we say that it is an *exogenous* pattern. On the other hand, a subworld can be a pattern because it has many internal redundancies, in which case we say that it is an *endogenous* pattern. In order to make such distinction, we take into account the compression rate of the subworld as if it were isolated indeed.

Definition 4. Let w, w_1, w_2 be $n-$worlds such that $w = w_1 \cup w_2$. We say that w_1 is a *pattern* in w under compression strategy C if $\texttt{CompRate}(C, w) < \texttt{CompRate}(C, w_2)$.

Besides, if $\texttt{CompRate}(C, w_1) \geq \texttt{CompRate}(C, w_2)$ then we say that w_1 is an *exogenous* pattern; if $\texttt{CompRate}(C, w_1) < \texttt{CompRate}(C, w_2)$ then we say that w_1 is an *endogenous* pattern. □

The previous definition does not imply that patterns have a *low* entropy level, but that their entropy is low in *relation* with the entropy of the world where they exist. In fact, even if a pattern is endogenous, it is not guaranteed that its entropy level is low, as the rate is measured by taking into account the world it belongs to.

Patterns can be nested, that is, we can find life inside living entities. By introducing this concept, we can manage notions such as cooperative living subentities constituting global living entities. We assume that the same compression strategies are considered at both nesting levels.

Definition 5. Let w_1 be a pattern in w under compression strategy C and let w_2 be a pattern in w_1 under compression strategy C. Then, we say that w_2 is a *subpattern* of w_1. □

Let us consider the notions of entropy and life. Intuitively, and following the ideas shown in [9], a living creature is a structure that maintains low entropy inside it, while increasing the entropy of the environment surrounding it. Thus, in order to decide whether a pattern is a living pattern or not, we have to compare the entropy of the pattern with that of its surroundings. We define the entropy of a subworld as the ratio between the entropy of that subworld and that of the world it belongs to. Next we introduce a notion of pattern which is parameterized by an entropy threshold. It allows us to compare patterns in terms of their relative level of order with respect to their world.

Definition 6. Let w, w' be two worlds such that w' is a subworld of w. The *entropy level* of w' in w under compression strategy C, denoted by $\texttt{Entropy}(C, w', w)$, is given by $\frac{\texttt{CompRate}(C,w')}{\texttt{CompRate}(C,w)}$.

We say that w' is an $\alpha-$*ordered pattern* of w under compression strategy C if w' is a pattern of w under compression strategy C and $\texttt{Entropy}(C, w', w) < \alpha$ for a given constant α. □

Even if a pattern is ordered inside its world, this does not imply that the pattern is a living pattern. We must also take into account that living entities must *increase* the entropy around them. Thus, we need to detect if there exists an *evolution* towards higher entropy. Reasoning about how some parameter evolves requires to identify a dimension of the world (or linear combination of them) as the *time* dimension, i.e. we have to define what is the direction of the evolution. Next we define sequences of increasing entropy. We say that a pattern is *alive* if it keeps a low entropy level along the evolution of its world and, simultaneously, the entropy level of its world increases along time.

Definition 7. Let $w_1, \ldots, w_n, w$ be worlds such that $w = \bigcup_{i=1}^{n} w_i$. We say that the sequence $w_1 \cdots w_n$ is an *evolution of entropy* under compression strategy C if for any i, j with $1 \leq i < j \leq n$ we have $\texttt{CompRate}(C, w_i) < \texttt{CompRate}(C, w_j)$.

Let w, w', w'' be worlds such that $w = w' \cup w''$ and w' is an $\alpha-$ordered pattern of w under compression strategy C. Let $w_1 \cdots w_n$ with $w = \bigcup_{i=1}^{n} w_i$ be an evolution of entropy under compression strategy C. We say that w' is an $\alpha-$*living pattern* across $w_1 \cdots w_n$ under C if there exist two sequences $w'_1 \cdots w'_n$ with $w' = \bigcup_{i=1}^{n} w'_i$ and $w''_1 \cdots w''_n$ with $w'' = \bigcup_{i=1}^{n} w''_i$, such that $w''_1 \cdots w''_n$ is an evolution of entropy under C and for all $1 \leq i \leq n$ we have that $w_i = w'_i \cup w''_i$ and w'_i is an $\alpha-$ordered pattern of w_i under C. □

It is worth to point out that $w'_1 \cdots w'_n$ (that is, the sequence representing the evolution of the living entity) could also be an evolution of entropy. That is, the internal entropy of an alive creature could also be increasing, provided that it is still a pattern inside its world. Intuitively, this implies that the tendency of the world towards chaos must be *faster* than the tendency of the living entity itself.

As we said before, the evolution of the entropy is not constrained to follow a specific direction. Since there are different ways to split a world into scenes, there exist several possible interpretations of time, and all notions depend on this choice. This increases the generality of the proposed framework. Let us remark that we are dealing with *information*, so our definition must be independent of the possible transformations being applied to such information. For instance, let us suppose that the world represents a video stream. Each temporal frame of the video could be located in a different part of the x axis of the information stream (e.g., a file). The evolution of the video over time is codified by locating each frame in a specific physical area of the stream. Hence, a flexible way to identify the time dimension must be provided.

It is worth to point out that the previous definition does take into account one of the factors considered critical for identifying life in terms of the Maximum Entropy Production Principle [9]. According to this principle, living creatures *generate* entropy in their surroundings. That is, they are the *reason* of the increment of entropy. In our approach, we detect life by just *observing* information, that is, we do not interact with it. Hence, we do not have the capability of changing the observing environment, which would allow us to check an alternative scenario where the creature does not exist. This would allow us to compare the evolution of the entropy in both cases, which is required to determine if the existence of the creature causes it. Studying the case where it is possible to interact with the analyzed environment is out of the scope of this paper and is left as future research.

Once we have proposed our notion of life, we can use it to define some higher level concepts. Next we consider the notion of *subliving patterns*. A subliving pattern is a living entity inside another living entity. In particular, the world of a subliving entity is the living entity it belongs to.

Definition 8. Let w' be an $\alpha-$living pattern across $w_1 \cdots w_n$ under the compression strategy C, and let w'' be an $\alpha-$living pattern across $w'_1 \cdots w'_n$ under the compression strategy C such that $w' = \bigcup_{i=1}^{n} w'_i$ and for all $1 \leq i \leq n$ we have $w'_i \subseteq w_i$. Then, we say that w'' is an $\alpha-$*subliving pattern* of w' across $w_1 \cdots w_n$ under C. We say that w' is a *fully* $\alpha-$*living pattern* across $w_1 \cdots w_n$ if there exist m worlds $w'_1, \ldots, w'_m$ $(m \geq 2)$ such that $w' = \bigcup_{i=1}^{m} w'_i$ and for all $1 \leq i \leq m$ we have that w'_i is an $\alpha-$subliving pattern of w' across the evolution $w_1 \cdots w_n$. □

As stated in [13], the *biological diversity* is the variety and variability among living organisms and the ecological environments in which they occur. Thus, the *diversity* can be defined as the number of different items and their relative frequency. In order to calculate the biodiversity of a world in our framework, we have to consider the life existing in it. Nevertheless, since diversity is required, the biodiversity does not increase by considering very similar living beings. On the contrary, the diversity is high only if it is possible to find a subset of the world such that its diversity is high. This subset should be defined in such a way that its members are *canonical representatives* of the different models of life appearing in the ecosystem. Then, the biodiversity will be calculated by considering two factors: The internal diversity of the subset (which indicates that present models are different among them) and its size (which indicates the amount of diverse life in the ecosystem). Hence, we calculate the biodiversity of a world by selecting the set that maximizes both factors together.

Definition 9. Let $w_1 \cdots w_n$ be an evolution of entropy under C, and let $w = \bigcup_{i=1}^{n} w_i$. The $\alpha-$*biodiversity* of $w_1 \cdots w_n$ under the compression strategy C is defined as:

$$\max \left\{ \frac{|\, S_{w'} \,|}{|\, S_w \,|} \cdot \texttt{CompRate}(C, w') \,\middle|\; w' = \bigcup_{i=1}^{m} w'_i \;\wedge\; \forall\, 1 \leq i \leq m : w'_i \in L \right\}$$

where L denotes the set of all $\alpha-$living patterns across $w_1 \cdots w_n$ under the compression strategy C. □

In the previous definition, we search the set of living beings such that, considering this set as a whole, the compression rate is the highest (which indicates that the diversity is high). At the same time, we search for the set whose size is as closer as possible to the size of the whole world (which indicates that the amount of diverse life is high). The multiplication of both factors provides our measure of biodiversity. Let us remark that the biodiversity is monotonic non-decreasing with respect to α. This is because higher values of α increase the freedom to choose living patterns, which allows to maximize the biodiversity value. In particular, those sets we can consider with a lower α can also be selected with a higher one.

The proposed formal framework also allows to define the notions such as *births* and *deaths* for living entities. For the sake of clarity, in previous definitions we assumed that each alive entity is alive during the *whole* considered period (i.e., during the considered evolution of entropy). Nevertheless, we can extend the previous concepts to deal with a more general situation where creatures are born and die.

In the following definition we introduce the concepts of birth and death. Let us remark that both concepts are relative to the entropy level α required in each case to determine if patterns are alive.

Definition 10. Let $w_1 \cdots w_m$ be an evolution of entropy under the compression strategy C and let w' be an $\alpha-$living pattern across $w_k \cdots w_n$ under the compression strategy C, where $1 \leq k \leq n \leq m$. Finally, let $w = \bigcup_{i=k}^{n} w_i$.

We say that the $\alpha-$*birth date* of w' is w_k if there does not exist $w'' \subseteq w_{k-1}$ such that w'' is an $\alpha-$ordered pattern of w_{k-1} under the compression strategy C and $w' \cup w''$ is an $\alpha-$ordered pattern of $w_{k-1} \cup w$ under the compression strategy C.

We say that the $\alpha-$*death date* of w' is w_n if there does not exist $w'' \subseteq w_{n+1}$ such that w'' is an $\alpha-$ordered pattern of w_{n+1} under the compression strategy C and $w' \cup w''$ is an $\alpha-$ordered pattern of $w_{n+1} \cup w$ under the compression strategy C. □

The intuitive idea behind the dates of birth and death is that they are dates such that it is not possible to extend the life of the creature after its death or before its birth.

3 Experiments

In this section we present an example of the framework presented in this paper by using a classical software game. This game is *Snake*. Essentially, the goal of this game consists in making the snake to grow up as much as possible by eating all the food it finds in the world. The snake dies either if it crashes against a part of its own body or against one of the walls surrounding the world. In addition to the original rules of the classic game, we introduce a new concept that will be necessary to deal with the proposed notion of life: Rubbish. When the snake eats something, it randomly produces rubbish in the surroundings next to it. This simulates the degradation of the environment caused by life. Since our notion of life requires that the entropy of the environment grows along time, a kind of degradation will be required to find life in this system.

We represent an execution of this game by means of a *world*. According to Definitions 1, 2, and 3, we consider a 3-world (2 spacial dimensions plus the *time*) where the size of each spacial dimension is 512. A *frame* of the scene (that is, the information of the world for a specific time) is shown in Figure 3 (left). The food is shown in blue color and the snake is green. In the following, we will use w to represent a subworld denoting a single frame.

Living structures are images moving across the screen along time, so we must be able to systematically search for parts of the image to be considered as possible living structures. In order to do it, we present an algorithm that automatically considers different ways to split the screen into pieces of different size and assesses the suitability of each piece to denote a living structure. The main part of the algorithm is depicted in the adjacent figure. This heuristic greedy algorithm looks for a square whose size is a divisor of n, being n the length of each spatial dimension. The cost of the algorithm is $\mathcal{O}(log(n))$. Intuitively, the algorithm works as follows: First, we split the image into four quadrants.

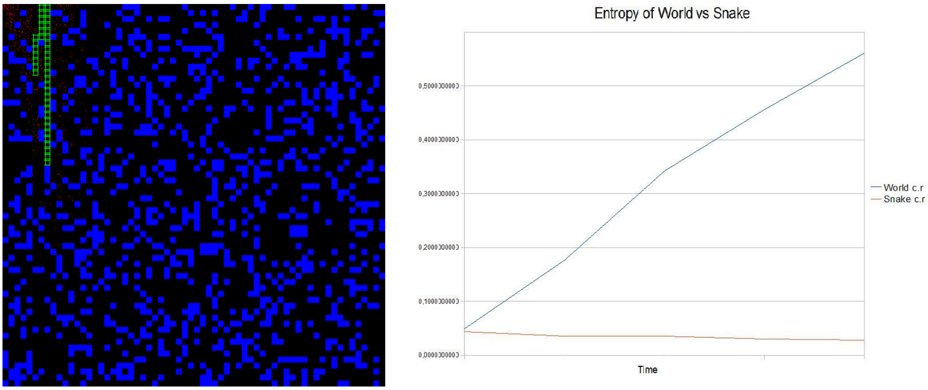

Fig. 1. A world representing an image (left) and the evolution of compression rate (c.r.) of the world and the snake (right)

We choose the quadrant where the conditions required to find life are best suited (that is, the square has low entropy but, at the same time, the entropy evolves along time in the rest of the frame). The function `BestComp` abstracts the criterion used to make this selection. Next, this quadrant is split again into four quadrants, and so on. When the process finishes, that is, when the minimal size is reached, the best square considered so far (regardless of its size) is identified. For the sake of clarity, some subsequent operations of the algorithm are not depicted in the figure. In order to *extend* the best square in some directions, other squares adjacent to it are considered. If the figure resulting by adding an adjacent square is better suited, then we take it as new best figure, and we repeat the same process for some additional turns. In this way, more complex forms (not just squares) can be formed. More formally, given a subworld w denoting a frame, the algorithm chooses two different subworlds w_1 and w_2, that is, subsets of w, fulfilling the following conditions. The first subworld, w_1, is the rectangular area chosen by applying the algorithm depicted in Figure 2. We define w_2 as the the rest of the frame, that is, we have $w = w_1 \cup w_2$ with $S_{w_1} \cap S_{w_2} = \emptyset$.

As we said before, we consider that the entropy of a subworld is the ratio between the compression rate of that subworld and that of the world it belongs to. So, our notion of entropy is a notion of *relative* order between a subworld and the world this subworld is inside. Following this idea, we perform an experiment to determine whether the snake should be considered an (artificial) alive creature according to the proposed notions. We use the JPEG compression algorithm to measure the entropy along time: Each individual frame is compressed by using this algorithm, and the evolution of resulting compression rates are considered. As we said before, other compression algorithms lead to different implicit definitions of what should be considered an ordered pattern and what should not.

In Figure 3 (right) we can observe the evolution of the entropy along the time of w_1 (the snake) and w (the world). Since the snake represents a simple repetitive pattern, the complexity of its JPEG codifications (that is, the *size* of the compressed images representing its frames along time) do not significantly increase along time. On the contrary,

input : An n-world represented by a Bitmap Matrix H of size $n \times n$.
output: A bitmap denoting a good life candidate within the world.

$n \leftarrow (n\ DIV\ 2) \times 2$;
$size \leftarrow \frac{n}{2}$;
$B \leftarrow$ MAXINT; $left \leftarrow 1$; $right \leftarrow n$; $up \leftarrow 1$; $down \leftarrow n$;
while (size ≥ 1) **do**
  $left_{new} \leftarrow left$;
  $right_{new} \leftarrow right$;
  $up_{new} \leftarrow up$;
  $down_{new} \leftarrow down$;
  if ($B \geq$ `BestComp`(H,$left$,$\frac{right}{2}$, up, $\frac{down}{2}$)) **then**
    $left_{new} \leftarrow left$;
    $right_{new} \leftarrow \frac{right}{2}$;
    $up_{new} \leftarrow up$;
    $down_{new} \leftarrow \frac{down}{2}$;
    $B \leftarrow$ `BestComp`(H,$left$,$\frac{right}{2}$, up, $\frac{down}{2}$);
  end
  if ($B \geq$ `BestComp`(H,$\frac{right}{2}$,$right$, up, $\frac{down}{2}$)) **then**
    $left_{new} \leftarrow \frac{right}{2}$;
    $right_{new} \leftarrow right$;
    $up_{new} \leftarrow up$;
    $down_{new} \leftarrow \frac{down}{2}$;
    $B \leftarrow$ `BestComp`(H,$\frac{right}{2}$,$right$, up, $\frac{down}{2}$);
  end
  ($\ldots continue$);
end

input : An n-world represented by a Bitmap Matrix H of size $n \times n$.
output: A bitmap denoting a good life candidate within the world.

$n \leftarrow (n\ DIV\ 2) \times 2$;
$size \leftarrow \frac{n}{2}$;
$B \leftarrow$ MAXINT; $left \leftarrow 1$; $right \leftarrow n$; $up \leftarrow 1$; $down \leftarrow n$;
while (size ≥ 1) **do**
  ($\ldots continue$);
  if ($B \geq$ `BestComp`(H,$\frac{right}{2}$,$right$, $\frac{down}{2}$, $down$)) **then**
    $left_{new} \leftarrow \frac{right}{2}$;
    $right_{new} \leftarrow right$;
    $up_{new} \leftarrow \frac{down}{2}$;
    $down_{new} \leftarrow down$;
    $B \leftarrow$ `BestComp`(H,$\frac{right}{2}$,$right$, $\frac{down}{2}$, $down$);
  end
  if ($B \geq$ `BestComp`(H,$left$,$\frac{right}{2}$, $\frac{down}{2}$, $down$)) **then**
    $left_{new} \leftarrow left$;
    $right_{new} \leftarrow \frac{right}{2}$;
    $up_{new} \leftarrow \frac{down}{2}$;
    $down_{new} \leftarrow down$;
  end
  $left \leftarrow left_{new}$;
  $right \leftarrow right_{new}$;
  $up \leftarrow up_{new}$;
  $down \leftarrow down_{new}$;
  $size \leftarrow \frac{size}{2}$;
end

Fig. 2. Searching good life candidate in an image

the JPEG codifications of frames representing the world become longer as time passes. Since the snake increases the rubbish every time it eats, the amount of rubbish increases along time, and representing this information in the compressed format requires more bits. Since the entropy of the snake remains low and the entropy of its world increases along time, we can conclude that the snake represents an alive pattern according to the notions presented in previous sections. Let us note that, as we have already commented before, we need that the amount of rubbish increases as the snake eats and grows. Otherwise, the world would reduce its entropy along time. Let us note that the compression rate of an *empty* world is better (uniformly colored areas are easier to compress). Thus, if the rubbish were not generated then the world of the snake would not be globally tend towards chaos according to the selected compression strategy.

4 Conclusions and Future Work

In this paper we have presented a formal framework to identify living entities inside an abstract information environment. Following the Maximum Entropy Production Principle, the proposed method is based on the analysis of the entropy of the components of the system. More precisely, we have compared the entropy of entities with the evolution of the entropy of the world they belong to. The entropy is measured in terms of *compression rates*. This allows us to measure the order degree of some information in a computational environment. Other related concepts, including notions such as death, biodiversity, or biologic families, have been discussed. We have illustrated the proposed concepts with a toy example where a living entity is detected in a simple game execution. Since the compression rate degrades but, simultaneously, the analyzed pattern remains ordered along time, we have concluded that this entity constitutes an alive entity according to the Maximum Entropy Production Principle.

As future work, we want to apply the proposed formal framework to analyze the presence of life in classical artificial environments. In particular, we wish to compare our definition of Life with Class IV considered by [15] and the Lambda metric proposed by [7] in the specific context of Cellular Automata. Besides, we wish to define alternative life detection notions. Contrarily to the formal notions presented in this paper, which are based on the simple *observation* of the environment, we wish to consider an alternative framework where we could extract conclusions by *interacting* with the analyzed environment.

Acknowledgments. We would like to thank Manuel Núñez and Natalia López for their interesting suggestions, as well as the anonymous reviewers for their valuable comments.

References

1. Berlekamp, E.R., Conway, J.H., Guy, R.K.: Winning Ways for Your Mathematical Plays. Academic Press, London (1982)
2. Bruers, S.: A discussion on maximum entropy production and information theory. Journal of Physics A: Mathematical and Theoretical 40(27), 7441–7450 (2007)
3. Dewar, R.: Information theory explanation of the fluctuation theorem, maximum entropy production and self-organized criticality in non-equilibrium stationary states. Journal of Physics A: Mathematical and General 36(3), 631–641 (2003)
4. Huffman, D.A.: A method for the construction of minimum redundancy codes. Proceedings of the Institute of Radio Engineers 40(9), 1098–1101 (1952)
5. Kim, K.-J., Cho, S.-B.: A comprehensive overview of the applications of artificial life. Artificial Life 12(1), 153–182 (2006)
6. Kok, C.W.: Fast algorithms for computing discrete cosine transform. IEEE Transactions on Signal Processing 45, 757–760 (1997)
7. Langton, C.: Studying artificial life with cellular automata. Physica D 22, 120–149 (1986)
8. Langton, C.: Artificial Life: An Overview. MIT Press, Cambridge (1995)
9. Prigogine, I., Stengers, I.: Order Out of Chaos. Bantam Books (1984)
10. Rafiei, D., Mendelzon, A.: Efficient retrieval of similar time sequences using DFT. In: 5th International Conference on Foundations of Data Organization and Algorithms (FODO 1998) (November 1998)
11. Schneider, E.D., Kay, J.J.: Life as a manifestation of the second law of thermodynamics. Mathematical and Computer Modelling 19(6-8), 25–48 (1994)
12. Shannon, C.E.: A mathematical theory of communication. Bell Systems Technical Journal 27, 379–423 (1948)

13. Office of Technology Assessment U.S. Congress. Technologies to mantain biological diversity, OTA-F-330. U.S. Goverment Printing Office, Washington (March 1987)
14. Welch, T.A.: A technique for high-performance data compression. IEEE Computer 17(6), 8–19 (1984)
15. Wolfram, S.: Cellular automata. Los Alamos Science 9, 2–21 (1983)
16. Wolfram, S.: Cellular Automata and Complexity. Addison-Wesley, Reading (1994)