Computers & Graphics 50 (2015) 13–24

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Procedural bread making ☆

Rodrigo Baravalle [a,*], Gustavo Ariel Patow [b,1], Claudio Delrieux [c,2]

[a] CIFASIS-CONICET, Laboratory for System Dynamics and Signal Processing, Ocampo y Esmeralda, S2000EZP Rosario, Santa Fe, Argentina
[b] Grup de Grafics de Girona, Departament d'Informàtica i Matemàtica Aplicada, Universitat de Girona, Edifici P-IV, Campus Montilivi, E-17071 - Girona (Girona), Spain
[c] Department of Electrical Engineering and Computers, Universidad Nacional del Sur - IIIE-CONICET, Avenida Colón 80, Bahía Blanca 8000FTN, Argentina

ARTICLE INFO

Article history:
Received 6 February 2015
Received in revised form
9 May 2015
Accepted 10 May 2015
Available online 20 May 2015

Keywords:
Bread
Photorealism
Fractal
Procedural models

ABSTRACT

Accurate modeling and rendering of food, and in particular of bread and other baked edible stuff, have not received as much attention as other materials in the photorealistic rendering literature. In particular, bread turns out to be a structurally complex material, and the eye is very precise in spotting improper models, making adequate bread modeling a difficult task. In this paper we present an accurate computational bread making model that allows us to faithfully represent the geometrical structure and the appearance of bread through its making process. This is achieved by a careful simulation of the conditions during proving and baking to get realistically looking bread. Our results are successfully compared to real bread by both visual inspection and by a multifractal-based error metric.

1. Introduction

Since its very beginning, computer graphics was concerned with the achievement of photorealistic modeling and rendering of real-world scenes [1]. There have been large ongoing efforts to produce realistic results for a large variety of scenes, from natural to synthetic, and from natural landscapes to urban scenarios. This has resulted in spectacular models to represent almost every possible material, natural or man-made, from water to fire, and from clouds to concrete.

However, not much attention has been paid to the modeling and rendering of food and edible materials, and only a few efforts can be mentioned in the literature [2–5]. On the other hand, accurate modeling of bread and its baking process has attracted a lot of attention in food engineering (e.g., [6]). However, this research is not taken advantage of in the computer graphics community.

In this paper we aim to produce a flexible and realistic model of bread, performing an accurate simulation of the different stages that bread undergoes during its cooking process, in particular proving and baking. For that, we use state-of-the-art models from food engineering, simulating first the gas bubbles formation in the dough during proving, and then the heat and mass transfer processes during baking.

The computational modeling of these processes, together with the initial shaping, is encapsulated in a procedural pipeline that is inspired in real bread formation. The pipeline is flexible enough to model arbitrary instances of the material, both in (macroscopic) shape and in (mesoscopic) texture.

The computation can be interrupted to show the material at intermediate stages, for instance to slice the raw dough, or to see the appearance of the bread after partial cooking. Since the crust making process is extremely complex, and not yet fully understood, we added an ad hoc mechanism, inspired in real crust measurements, to simulate and render crusts after the baking process. The complete modeling process requires little or no supervision.

The final images of the crumb and crust of baked breads appear to be quite realistic. To phenomenologically validate the results, we compared slices of the volume texture induced by the bubble distributions with different real bread types using multifractal measures. As far as we could find in the literature, this is the first attempt to develop an automatic process that can be configured to produce realistic simulations of bread.

2. Previous work

Procedural modeling of geometry substantially reduces the need for artistic intervention in domains or situations where repetitive supervised action would turn out impractical, for instance in shaping cities [7], planets [8], buildings [9], and plants [10]. Some methods employ grammars to define mathematical descriptions that represent spatial relationships between primitives, for instance cubes,

☆This article was recommended for publication by Bedrich Benes.

* Corresponding author. Tel.: +54 341 215 6499; fax: +54 341 482 1771.
E-mail addresses: baravalle@cifasis-conicet.gov.ar (R. Baravalle), dagush@ima.udg.es (G.A. Patow), cad@uns.edu.ar (C. Delrieux).
[1] Tel.: +34 972 41 88 32; fax: +34 972 41 87 92.
[2] Tel.: +54 291 4595101lxt3381.

http://dx.doi.org/10.1016/j.cag.2015.05.003
0097-8493/

cylinders, or lines. The final structures usually arise using recursion over tokens in the grammar.

Although there was some initial research in computer graphics on bread crumb modeling and rendering [2,3], the overall process of bread making involves several stages that were not always well accounted for in the field. Earlier works applied physical baking models to certain bread types for rendering animations (e.g. [11]), but the bread crumb bubbles' geometric modeling issue was not considered.

Procedural bread modeling, on the other hand, is a multidisciplinary research subject. Literature in food engineering has various decades of ongoing research aimed at understanding the bread production process. Studies in this area show that the proving stage of bread making strongly determines the features present in bread crumbs, in particular the bubbles [12]. The interaction between the yeast and some nutrients present in the dough produces CO_2. Bubbles' radii and their spatial distributions show fractal-like structures, exhibiting a distinguishable statistical self-similarity at different measurement scales. Studies computed different fractal dimensions for these structures in certain bread types [13], suggesting uniform fractal bubble distributions. The bread baking modeling has also been a subject of significant research [14].

Procedural fractal representations for other materials gave rise to a wide variety of research interests, including disparate topics such as mountains [15], moon craters, and bubble size distributions in cheeses [16]. In addition, complex mathematical models represent the behavior and growing of several natural phenomena. In computer graphics, these models are one of the foundations used to model water and fluids [17,18]. Authors borrow complex differential models from other science fields and compute them using numerical techniques. In recent years, GPGPU technology [19] enabled the possibility of real-time or interactive frame rate computation and rendering of these numerical models.

Notwithstanding these breakthroughs, our final goal still presents several challenges. In addition to an accurate mesoscopic model (3D texture), bread crumb rendering requires an adequate representation of the light transport phenomena, including self-occlusion, self-shadowing, transmittance, and reflection, among others. Only a few publications propose to manage both the geometric and illumination models, and mostly stressing only artistic considerations [3]. Also, these authors did not disclose enough details of the modeling and rendering algorithms (probably due to intellectual property issues) and thus the results are hardly reproducible.

On the other hand, the artistic community usually produces realistic bread rendering results using photographs and defining geometries from them, using translucent materials and subsurface scattering effects[3] and other ad hoc considerations.[4] These solutions generate realistic outcomes, but the required processes are tedious and time demanding. Moreover, defining different bread models requires repeating the whole modeling afresh, making the process far from practical. This way of coping with bread modeling is rigid in nature, and therefore has several other drawbacks. For instance, it is not possible to get arbitrary slices of the resulting object.

3. A pipeline for bread modeling

In relation to the previous work mentioned above, we propose to unify and differentiate the key steps of bread making (proving and baking) to produce a physically inspired pipeline for procedural generation of bread crumb and crust materials. These processes have to be well understood before development of accurate modeling.

In this section we describe all the stages that lead to the final generation and rendering of a realistic bread material. First, we review the state-of-the-art in understanding the bread baking process and we briefly introduce some ancillary mathematical models that will be required for understanding the main computational developments that we will present in our work. Then, we present the modeling procedures that, together with the illumination model, give rise to the main results presented in this paper.

The whole process is comprised in a modeling pipeline (see Fig. 1), where all the stages that emulate bread making are processes operating on a (binary) voxelized space. The user can feed the pipeline with a 3D mesh model of their preference, or allow the system to provide standard shapes, usually found in bakeries. Then, this 3D model is voxelized into a volumetric texture in order to proceed. During the bread proving simulation, the user can change the voxelization of the dough (parameterizing the bubble distribution – amounts and radii of bubbles–), or, again, leave the default parameters that lead to a generic voxelized space and, latter on, to a generic appearance. The dough shape will be intersected with the voxelized geometry in the previous step (the external shape provided by the user) to obtain a final volumetric texture, with its interior filled with the bread material (see Section 3.2). Then a specific baking model is computed [20] to rise the dough while slightly twisting the volumetric texture (and the bubbles therein) according to the effects that baking produces in the bread making process. Finally we apply direct volume rendering (DVR) [21] to the final voxelization, obtaining realistic images of the resulting bread.

3.1. Model voxelization

Users can generate arbitrary 3D bread shapes using our pipeline. The first step voxelizes a user-provided mesh with the open source `binvox` utility[5] [22]. The voxelization generates a binary volumetric texture, where 1 represents that the given voxel is inside the geometry, and 0 means that the voxel is outside. In the proving simulation step (next subsection), we generate the material that lies inside this voxelized model. This allows us to generate arbitrary bread types such as *baguettes*, *sliced* breads, or fancy shapes.

Once the user-provided geometry is converted into a volumetric texture M, we generate a second volumetric texture of the same size that contains, in each voxel, the result of computing the distance to the closest boundary voxel in the original volumetric texture [23] as follows. Given a binary volumetric texture M, the algorithm evaluates the new volumetric texture DF_M as

$$DF_M[i,j,k] = \min\{\delta([i,j,k],[i',j',k']) : M[i',j',k'] = 0\},$$

where $[i,j,k]$ and $[i',j',k']$ are cells in the respective textures representing the (i,j,k) and (i',j',k') spatial positions, and δ is the function that computes the distance between two cells, computed as either the Manhattan or the Euclidean distances. Entries far from the object boundaries get higher values than closer ones. This new volumetric texture acts as a discrete distance field, which will be required later at different stages of the pipeline.

3.2. Proving simulation

As stated above, the actual volume texture of the bread dough is due to bubble patterns, which in turn are the result of complex processes, including chemical reactions and physical deformations. The proving step accounts for the free bubble growth produced by living yeast in the dough. Then, human intervention deforms

[3] http://www.blenderguru.com/tutorials/how-to-create-realistic-bread

[4] http://design.tutsplus.com/tutorials/create-a-realistic-loaf-of-bread-in-photoshop–psd-10555

[5] http://www.cs.princeton.edu/min/binvox/

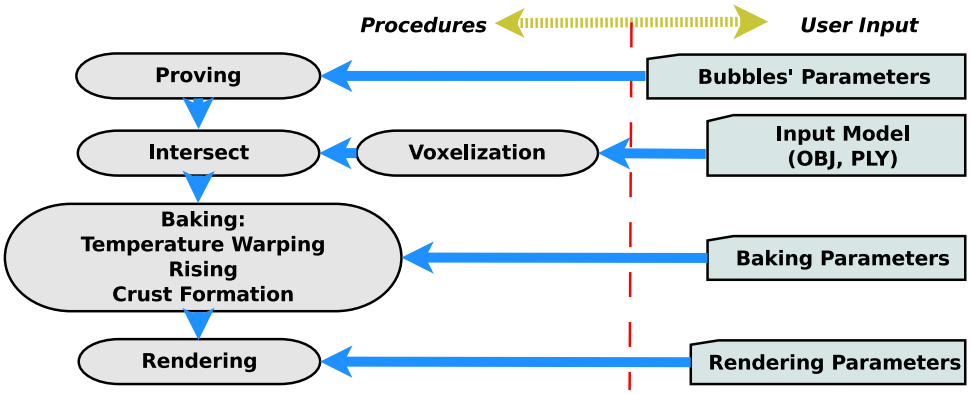

Fig. 1. Pipeline for synthetic bread making. The automatic procedures appear in rounded shapes (left) while the user inputs are shown in rectangular containers (right). The image shows that the user has control on the global shape, local bubbles, global and local deformations during baking, and colors for rendering.

dough shapes in several ways, and the baking process produces the final bubble shapes. Phenomenological studies of bubble distributions employ X-ray tomography devices and image feature extraction [12,13,24]. To obtain a similar structure variability, we generate bubble distributions with a fractal-based model inspired in the cheese and moon crater distributions proposed in [16], and we further validate it using the sandbox multifractal spectrum, as will be explained later in detail.

The volume texture of the dough is procedurally created in a separate 3D volumetric texture, in a similar way as in the previous subsection. Each voxel is initially set to 1, meaning that there are no bubbles so far. The process begins by subtracting randomly positioned spheres of radius r_{min} to this structure (by setting the respective cells to 0). Then, larger spheres are subtracted, also at random positions, up to a maximum radius r_{max}. The relationship between the amount $N(r)$ of spheres to be subtracted to the material at each step, and their respective radii r, is given by the following fractal law:

$$N(r) = \frac{k}{r^d},$$

where d is the fractal exponent that models the likelihood of occurrence of spheres in relation with their radii, and k controls the amount of actual spheres at each radius. Both parameters are sufficient to model a wide range of volume textures in general, and bread 3D textures in particular.

Fig. 2 shows a 2D slice of this model. Even though the resulting bubble shapes are not completely realistic (for instance, the spherical nature of the bubbles is apparent and artificial), the results show high resemblance in size and distribution with real bread binarizations (see [12]). During baking, the resulting volumetric texture will undergo geometric deformations so that the final texture will more accurately resemble actual bread. Quite related to this, the issue of finding adequate parameter values to obtain features similar to real bread crumbs will be further discussed in the validation section.

The volumetric texture P produced by this proving simulation must then be constrained to the interior of the user-provided geometry. For this, a naïve solution would be to intersect the two volumetric textures by means of a simple masking $P_1 = P \times M$, where P_1 is the resulting volumetric texture (i.e., the dough after being shaped to the user-provided geometry M). Fig. 3 shows some results of this naïve procedure.

Even though these results might appear satisfactory, a careful examination shows that bubbles are seldom present close to the exterior surface of real bread due to complex processes during proving (the exterior of the dough gets dehydrated and therefore the yeast is unable to form CO_2 bubbles). This process is not fully

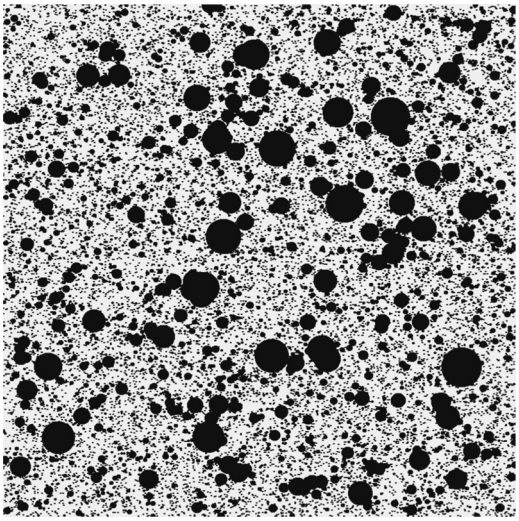

Fig. 2. Fractal bread proving simulation.

understood in the food engineering literature. To account for it in our model, the discrete distance field DF_M computed at the previous stage is used to disallow the generation of bubbles close to the geometry boundary. Bubble generation, then, is constrained to occur only when the outermost part of the bubble is within a given threshold distance to the external boundary of the shape.

Figs. 4 and 5 show the result of limiting the bubble interactions to the inner region of the original voxelized geometry. These images show that the approach also allows us to produce realistic arbitrary unbaked doughs. Note that our model is flexible enough to render the material at any stage of the pipeline, and also to arbitrarily slice or intersect the model.

3.3. Baking

Adequate physical models of the bread baking process should be able to accurately model the heat and mass transfer in dough. Full 3D bread baking models may lead to precise solutions to the temperature's distribution in dough during baking. However, these models are usually very complex to implement and have high computational costs. In addition, only the most complex models take into account crust formation, but usually using overly simplified assumptions. This is due to the fact that a formal crust definition is currently missing [25]. Dough rising constitutes another visible phenomenon and modeling its coupling with heat and mass transfer results in a very complex model [26]. In our pipeline we simulate dough rising and bubble warping in one step.

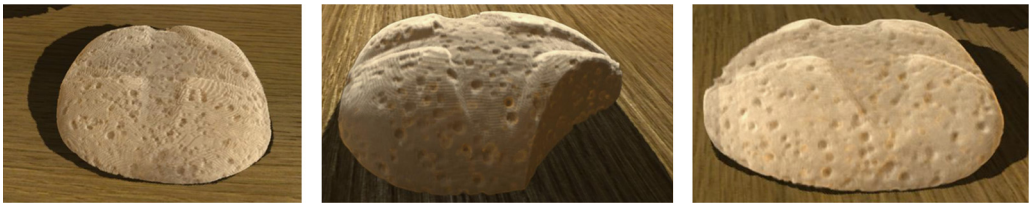

Fig. 3. Naïve geometry and bubbles intersection. Bubbles are unrealistically visible on the dough surface.

Fig. 4. Proved bread before baking.

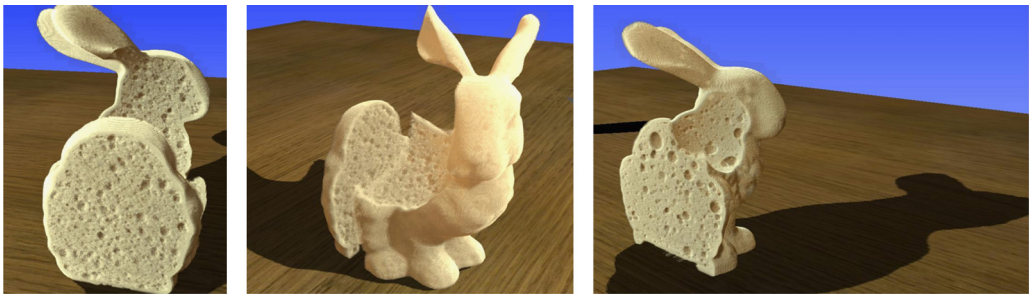

Fig. 5. Proved bunny-shaped bread before baking.

3.3.1. Temperature warping

The most relevant results suggest that a 1D model could suffice for most purposes. For instance, Purlis [6] models bread baking as a 1D problem, representing bread as an infinite cylinder. Other frameworks also assume only one radial coordinate, resulting also in 1D models [20,27]. These works show that employing a simple 1D representation produces almost identical results on the bubble deformation, as compared to the ones using higher dimensions.

The numerical simulation that we implemented is based on the finite-differences framework proposed by Powathil [20] and Thorvaldsson and Janestead [27]. It consists of a set of three coupled equations describing heat transfer, water vapor diffusion and liquid water diffusion. In our algorithm, we require only the computed temperatures (T) as the input for the next stages. Eq. (1) models heat transfer in bread dough, accounting for energy balance and water evaporation due to temperature [27]:

$$\frac{\partial T}{\partial t}=\frac{1}{\rho C_p}\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\lambda}{C_p}\frac{\partial W}{\partial t}+\frac{\lambda W}{\rho C_p}\frac{\partial \rho}{\partial t}, \tag{1}$$

where T is temperature, x is the radial coordinate, t is time, C_p is the specific heat, ρ is density, k is thermal conductivity, λ is the latent heat of evaporation of water, and $W(x,t)$ is the liquid water content. The initial conditions

$$T(x,0)=T_0(x), \quad 0\le x\le L/2,$$

and the boundary conditions (continuity and smoothness) define the model:

$$\left(\frac{\partial T}{\partial x}\right)_{x=L/2}=0, \quad t>0$$

$$-k\left(\frac{\partial T}{\partial x}\right)_{x=0}=h_r(T_r-T_s)+h_c(T_{air}-T_s)-\lambda\rho D_w\left(\frac{\partial W}{\partial x}\right)_{x=0}$$

where h_r and h_c are subterms of the heat transfer coefficient ($h=h_r+h_c$), T_{air}, T_s, T_r are the temperatures in the surrounding air, at the surface of the bread and at the radiation source, respectively, L is the bread height ($x=L/2$ is the bread center and $x=0$ is the bread boundary), D_w is liquid water diffusivity, and T_0 is the initial temperature. Temperatures are expressed in Kelvin (K). The model presents similar equations for water vapor diffusion (W) and liquid water diffusion (V). Further details of this model can be found in [27]. We use this model to obtain an array of temperatures during the baking process. These temperatures in turn will affect the bubble shapes and sizes.

The baking simulation sets an oven at a typical baking temperature (by default we take 210 °C) and discretizes time in intervals of $\Delta t=30$ s, and from the simulation we obtain an array *Temp* of N_{grid} temperature values. For numerical stability, we set $N_{grid}=32$ and we interpolate the temperatures to obtain higher resolutions (N_{int}).

The array we obtained from baking has decreasing temperatures from its boundary (*Temp*[0]), where we can observe the highest temperature, to its center (*Temp*[$L/2$]), which has the lowest temperature (heat transfer flows from the boundary to the center). Then, we can use the previously computed discrete distance field DF_M in conjunction with the *Temp* array to map distances to temperatures. This allows us to define temperatures in the volume in a way that is compatible with a 3D baking simulation. Based on these considerations, we map the temperatures array *Temp* into 3D coordinates using the following relationship:

$$R_{vol}[x,y,z]=Temp[round(DF_M[x,y,z])],$$

where R_{vol} is the resulting volumetric texture, and x, y and z are 3D coordinates in R_{vol}. When $DF_M[x,y,z]=0$, the 3D position lies at the exterior of the geometry, and *Temp*[0] is mapped to $R_{vol}[x,y,z]$. When $DF[x,y,z]>0$, a lower temperature is mapped to R_{vol}, since the position lies in the geometry's interior. Fig. 6 shows slices of the result of this mapping (R_{vol}), using temperatures from an arbitrary baking time step in the three Cartesian planes. The images are almost identical to what will be obtained with a 3D simulation [28].

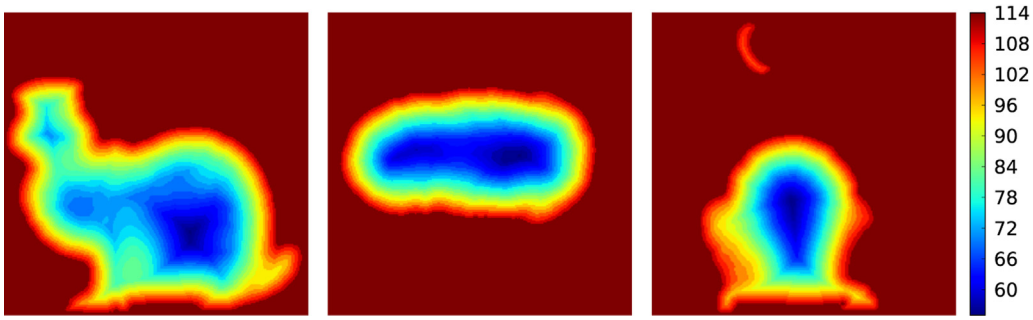

Fig. 6. Resulting temperatures from the baking simulation, mapped onto a bunny-shaped 3D volume. The images show that temperature distribution results similar to a complete 3D simulation.

Fig. 7. Two dimensional cuts showing bubbles after temperatures warping over different shapes.

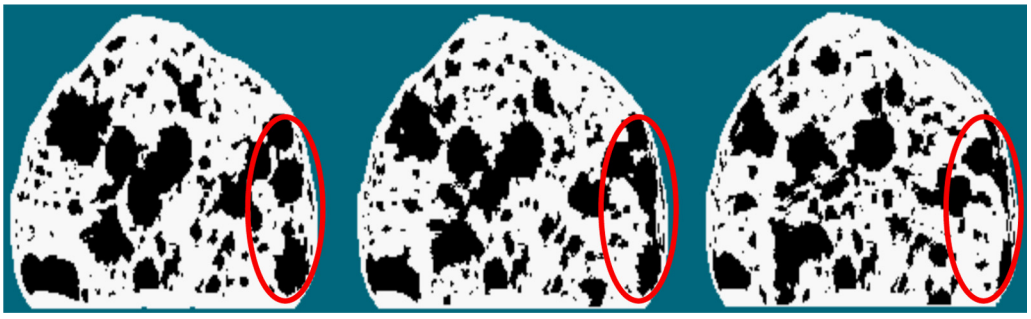

Fig. 8. Effect of the parameter p. From left to right p is 5, 10, and 15, respectively. Bubbles are realistically pushed and stretched towards the crust.

After a number of simulation steps (up to $t=20$ in our case), we compute the gradient vector g of R_{vol} [29], and we smooth it using a Gaussian kernel. We use a Gaussian smoothed version (g') to warp the original volume by modifying the sampling coordinates in the following way:

$$[x,y,z] = [u,v,w] + pg'[u,v,w],$$

where $[u,v,w]$ is the original voxel entry, $[x,y,z]$ the warped voxel entry, p a real positive valued parameter that denotes intensity (it controls the baking effect on bubbles) g' the smoothed Gaussian version of the original gradient g, and $g'[u,v,w]$ is the vector gradient at the entry $[u,v,w]$. Fig. 7 shows 2D slices with the process described applied to different bread shapes.

The parameter p can be used to synthesize different bread crumb appearances, from non-baked to highly deformed bubbles (see Fig. 8). When we increment p, we force the bubbles to follow more tightly the bread outer shape. The image also shows that the method more intensely deforms bubbles closer to the surface. Also, the image clearly shows the stretching of circular bubbles close to the right bread boundary.

3.3.2. Dough rise

Rising constitutes one of the most visible phenomena during bread baking. Temperature increase causes CO_2 to be released in dough bubbles, resulting in a net volume increase. Bubble volume changes are responsible for bread rising during baking. Studies show that, in a typical scenario, the bread dough grows to a volume that approximately doubles its original size [30], depending on oven conditions.

In our model, this process can be simulated using bubble radii distribution information from proving. To compute this information, we create an ancillary volumetric texture P storing the bubble radii at each entry:

$$P[x,y,z] = \max\{r\},$$

where r is the radius of a bubble that occur at the voxel entry $[x,y,z]$. The max term in the computation retains the maximum radius occurring at $[x,y,z]$. To obtain smooth radii transitions, a Gaussian filter is applied to P. This volumetric texture allows us to define several useful functions that simulate plausible approximations of dough rise.

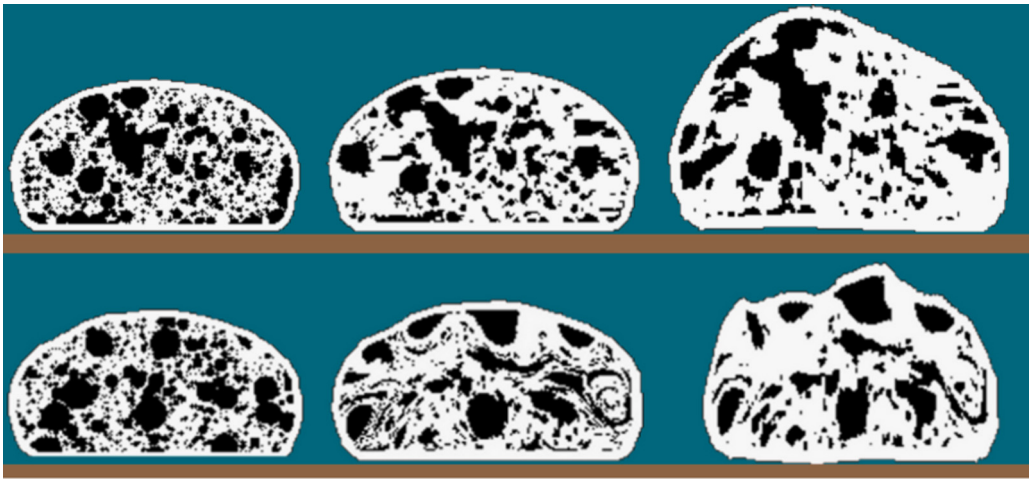

Fig. 9. Examples of dough rising, using different S values. The image shows the baking effect on the overall dough shape and on particular bubbles. Different bubble growth and bubble coalescence is observed.

The original volume will be warped with the values from P to obtain different growing ratios for different bubble radii. For this, we set the p value from the previous section with the P value for each entry. We warp the embedding coordinate system as follows:

$$[r,s,t] = [u,v,w] + P[u,v,w]g'[u,v,w],$$

where $[u,v,w]$ is the original voxel entry and $[r,s,t]$ is the warped voxel entry.

The actual rising is accomplished by means of local scaling. We deform the auxiliary volumetric texture using a function that scales it using a real valued parameter $S > 1$, and the bubble radius at the corresponding entry, producing the rising doughs:

$$[x,y,z] = [r,s,t]SP[r,s,t], \quad (2)$$

where $[x,y,z]$ is the final voxel entry in the dough after the warping induced by the rising.

Fig. 9 shows examples of doughs before and after rising. The image also shows different growths for different bubbles. In addition, we can observe that coalescence is produced on growing bubbles. The differences in growth can be randomized to obtain bubble growth variability.

3.4. Crust formation

The baking model we introduced does not include precise details of the crust formation. In these equations, the crust is assumed to be produced on the surface and at certain temperature, but this assumption is not exactly what actually happens in the real baking process. The food engineering literature defines the crust as an interface that appears between the dough and the surrounding air. Its main characteristics include the formation of a thicker material at the dough surface, and color differentiation, but the precise details surrounding the crust formation and appearance are still far from known.

The crust is mostly determined by an evaporation front which is produced by the temperatures in the dough [31]. Therefore, we can obtain crust positions using the distance field, since we have defined a relationship between temperature and distance to the surface. We use a distance parameter to obtain different thicknesses for the crust, getting different bread appearances.

In addition, we use colors measured from real baking [32] to set the appearance of the crust. We employ the L channel in the CIELab color space [33] to control baking histories at each crust voxel. The L channel determines luminance, while the a and b channels determine the chromaticity. In [32], it was stated that values in the L channel approximately ranges from 90 (unbaked) to 40 (burned). The chromaticity values for the channels were chosen using colors from real bread images. Using this information, we define a range of *CIELab* colors, smoothly ranging from unbaked to fully baked.

Several artistic considerations can be employed to distribute the defined colors over the crust. For instance, bubble proving information can be used (radius of the closest bubble), or even procedural noises (e.g., Perlin). At each crust position, the values obtained from these functions are used to choose a color from the previously defined colors.

In our case, we define the *crust density* at each position in the volumetric crust texture as N_v/W_{size}, where N_v is the number of voxels with value 1 in a window surrounding the position, and W_{size} is the voxel count in that window. We use this value to modulate the L channel in the crust color, at each position in the volumetric crust texture, obtaining plausible crust appearances. We based our choice in the observation that in places where there is fewer mass (for instance, corners in the crust), the baking effect is more noticeable. The size of the window can be used to obtain variability in the crust appearance.

3.5. Rendering

We applied Direct Volume Rendering (DVR) [21,34,35] to obtain the final images. DVR approximates the light transport equation by throwing rays from a virtual camera into a volume defined by a discrete 3D density function, accumulating density information and providing a 2D representation. DVR samples the density function along the ray to approximate the effect of different light phenomena such as extinction, transmittance and scattering, among others.

The choice of DVR over other state-of-the-art methods such as Ray Tracing [36,37], Path Tracing [38] and Photon Mapping [39], is mostly due to the fact that these methods are computationally expensive and require a detailed object mesh.

4. Results

In this section we present the results of our procedural bread making pipeline. Figs. 10–15 show DVR-rendered images of 3D volumes we obtained in this work. Figs. 10 and 11 highlight the differences in image quality between a low (256^3) and high (512^3) resolution versions respectively. Fig. 12 shows high resolution versions of other voxelized geometries. Fig. 13 shows the same crust with different baking histories. We achieve these effects by setting a

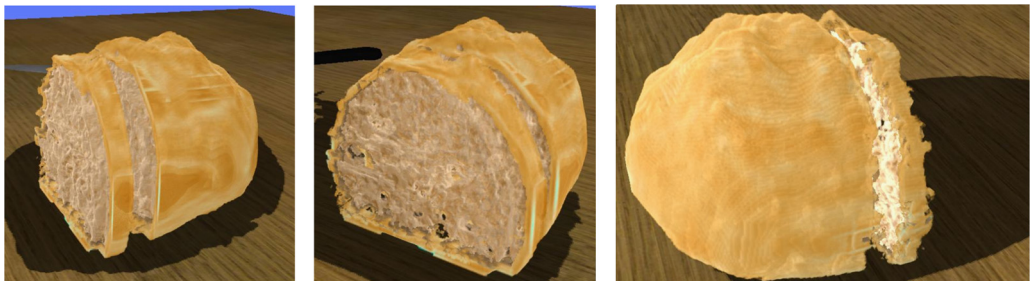

Fig. 10. Sliced breads after baking, using a volumetric texture of dimensions 256^3. The images show that crumb and crust are taken into account in our approach. The rising effect is appreciated.

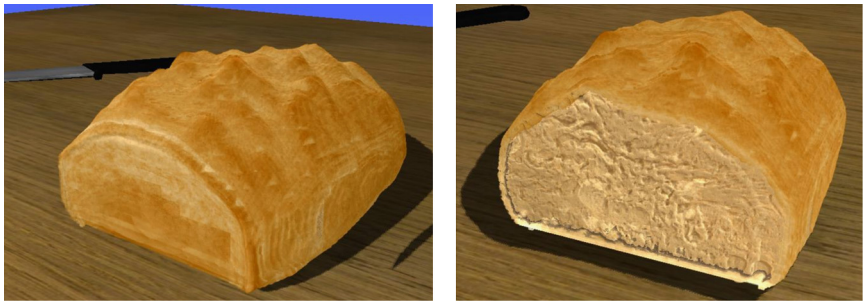

Fig. 11. Sliced breads after baking, using a texture of dimensions 512^3. Finer details can be seen in the crumb. Also, the *S* parameter was chosen with a random behavior at each point, producing visible spikes in the crust. The crust color slightly differs from the previous image.

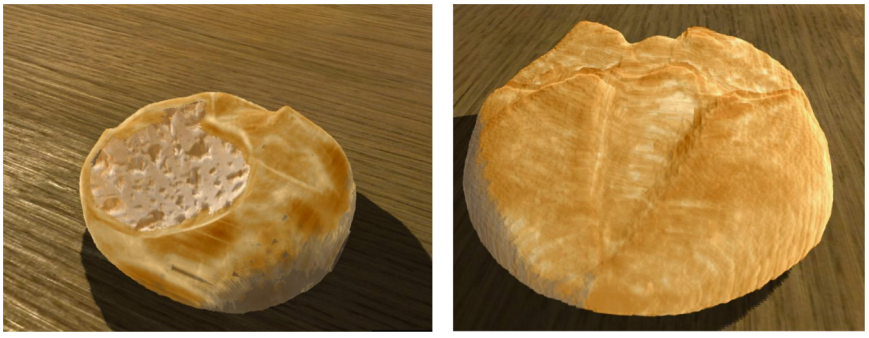

Fig. 12. Breads after baking. This image shows different 3D geometries and crust appearances.

Fig. 13. Crust colors tuned to show different baking histories. From left to right, the maximum allowed value for the *L* channel is decremented.

Fig. 14. Breads after baking showing different deformation parameters. The *S* parameter is incremented from left to right.

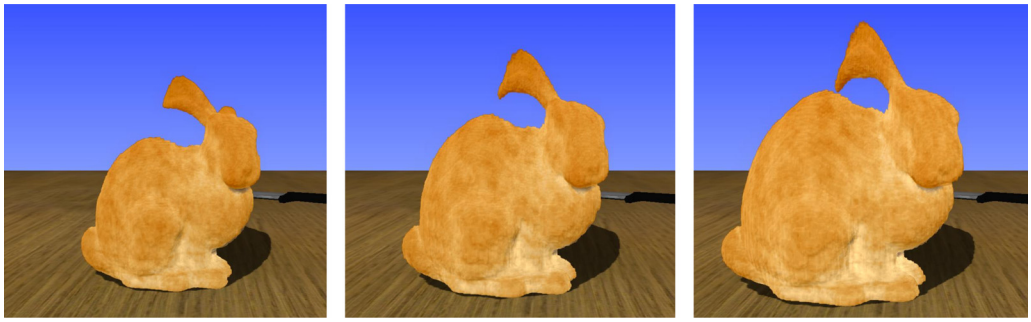

Fig. 15. Bunny-shaped breads after baking showing different deformation parameters. The S parameter is incremented from left to right. Images appear realistic even for unrealistic dough models.

different maximum L value in the *CIELab* color space for each image. Figs. 14 and 15 show the effect of setting different S values in Eq. (2), observing different rising outcomes. The images show that the method produces realistic baking results even for an unrealistic bunny-shaped dough.

Cuts and slices can be easily produced by setting to 0 different regions in the volume after baking and before rendering.

4.1. Computing times

We used Python[6] and Cython[7] for the implementation. In Table 1 we show typical times for the different processes in the bread making pipeline. In addition, rendering times with DVR are around 0.3 s, achieving interactive framerates.

It is worth to mention that most of the baking time is dedicated to the gradient computation and not the baking itself. Our code is currently not fully optimized, so better performance can be achieved with a more careful design at each stage. The user can generate previews using lower resolution matrices, avoiding high computing times when testing different bubble distributions and dough shapes.

Table 1
Typical computing times in our pipeline expressed in seconds.

Voxelization resolution	256^3	384^3	512^3
Proving	3.62	12.07	31.29
Intersection	8	10.27	14.97
Distance field	7	23.73	56
Baking	49.81	144.69	313.7

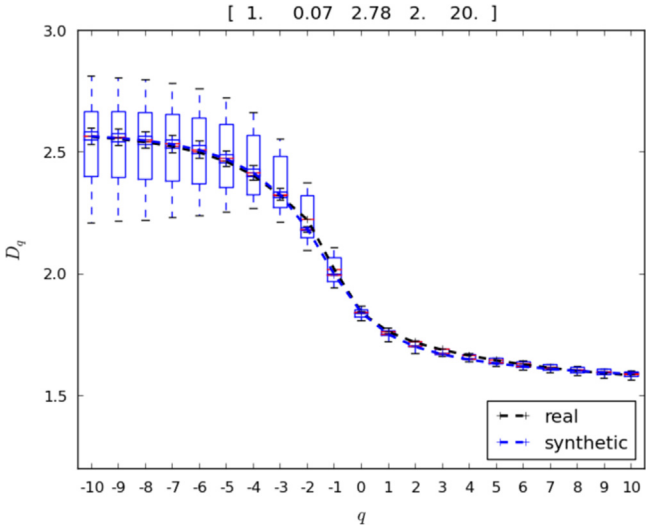

Fig. 16. Best fitting parameters for the *baguette* bread type. The total error in medians is ~0.21.

5. Model validation

This section is included for the sole purpose of having a more objective assessment of our model's visual quality, apart from what subjective inspection may indicate. We validated our model using multifractal features, in particular the multifractal spectrum method, that is widely used for texture analysis in computer vision and pattern analysis. Fractal Dimensions (FD) can be used to characterize key image features using just a few parameters. Different FDs capture different features, e.g., porosity and rugosity. Multifractal theory has also been used for image feature description, but not much in bread characterization. A few previous studies show fractal bread characterizations using several FDs [13,40]. There are two main classes of multifractal spectra: generalized multifractal dimensions (D_q) and Lipschitz–Hölder exponents ($f(\alpha)$). The latter representation boosts the performance in classification tasks.

The generalized multifractal dimensions can be computed in several ways, of which the Sandbox multifractal method [41] aims to compute the dimensions using the mean value in a set of randomly distributed points belonging to the structure [42]. The generalized multifractal dimension of order q with the sandbox method is defined as

$$D^{sb}_{q\neq 1} = \frac{1}{q-1}\lim_{R\to 0}\frac{\ln\left\langle (M(R)/M_0)^{q-1}\right\rangle}{\ln(R/L)},$$

$$D^{sb}_{q=1} = \lim_{R\to 0}\frac{\left\langle \ln(M(R)/M_0)\right\rangle}{\ln(R/L)},$$

where M_0 is the background pixel count in the image binarization and $M(R)$ is the foreground pixel count within a circle of radius R centered at a given foreground point in the structure. When $q\neq 1$, D^{sb}_q is computed as the limit of the slope of the linear fit of the values $\ln(R/L)$ vs. $\ln\left\langle (M(R)/M_0)^{q-1}\right\rangle$, for R in $[R_{min}, R_{max}]$, where $\langle\cdot\rangle$ denotes mean value over sampled points. The procedure is similar when $q=1$. The sandbox multifractal spectrum, then, is produced by computing D^{sb}_q for different values of $q\in[Q_{\min}, Q_{\max}]$.

This kind of analysis is reported to be the most adequate for geometrical and texture feature analysis [13,40], so we applied it to 10 binarized scanned real bread crumb images of the *baguette* bread type, and 10 synthetic images produced using our pipeline,

[6] python.org
[7] cython.org

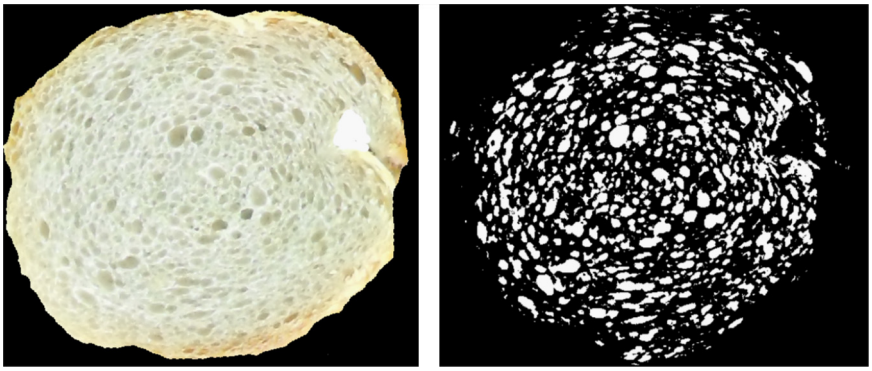

Fig. 17. Real *baguette* bread and binarization example.

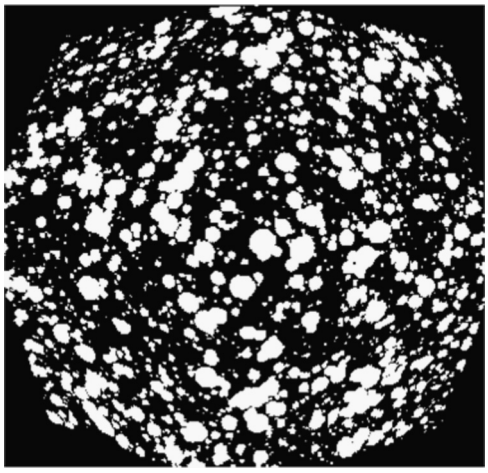

Fig. 18. Model distribution example showing similar multifractal features to the *baguette* real bread type.

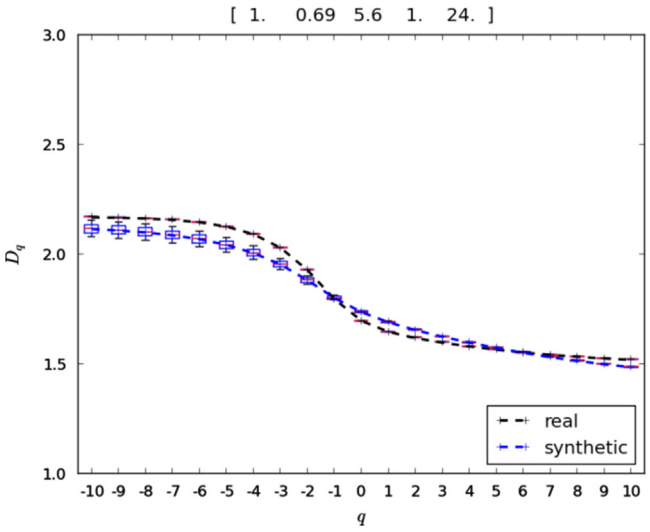

Fig. 19. Best fitting parameters for a home-baked bread type. The total error in medians is ~0.88.

after the baking step, obtaining 20 feature vectors. We then separated each class and we computed boxplots for the two classes. Actual bread crumbs were manually segmented to prevent errors from automatic segmentation as in [43]. The model has 5 parameters:

$$N(r) = \frac{k}{r^d},$$

$$r = r_{min} + step * j, j \in \left[0, \frac{r_{max}}{step}\right],$$

k, d, r_{min}, r_{max} and $step$ that controls bubble generation in the proving stage. To compute the sandbox spectrum of each image, we used 1000 different random points to grant stability in the resulting spectrum.

We implemented a partial automated search in parameter space. First, we manually set initial values for the parameters based on visual inspection of real and synthetic binarizations. Starting on those initial values, the method performs an automated search in parameter ranges centered on each initial value up to some threshold. The search exhaustively compares synthetic spectra with the real spectrum using different range steps for each parameter. The method performs the comparisons defining an error metric:

$$Error = \sum abs(mean_{real} - mean_{synthetic}).$$

The algorithm returns the lowest error found and the associated parameters.

For the baguette bread type, we found that the following parameters produce the smallest error:

$$k = 0.07\frac{R^3}{20},$$
$$d = 2.78,$$
$$r_{min} = 2,$$
$$r_{max} = 20.$$
$$step = 1,$$

where R is the resolution of the proving volume in each spatial coordinate (we chose $R=512$). The accumulated error in medians is ~ 0.21 meaning a mean error of $0.21/21 \sim 0.01$ in each dimension. Fig. 16 shows boxplots of real and synthetic breads with the medians for each dimension joined by dashed lines. When $q < 0$ dimensions have a higher dispersion, since the method approximates these dimensions less accurately. The figure shows almost identical spectra for real and synthetic breads. Figs. 17 and 18 show an example of real and synthetic binarizations for these parameters.

Also, we found parameters for a home-baked bread type using the same search:

$$k = 0.69\frac{R^3}{20},$$
$$d = 5.6,$$
$$r_{min} = 1,$$
$$r_{max} = 24.$$
$$step = 1,$$

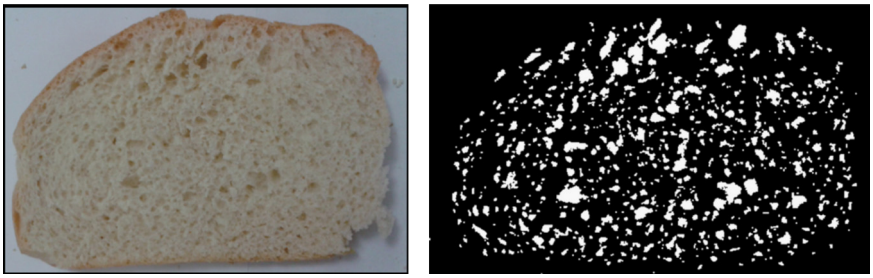

Fig. 20. Real home-baked bread and binarization example.

Fig. 19 shows boxplots of real and synthetic breads for this bread type. Figs. 20 and 21 show an example of real and synthetic binarizations for these parameters and bread type.

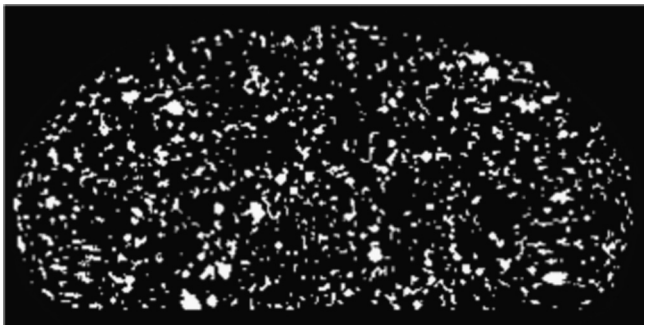

Fig. 21. Model distribution example showing similar multifractal features to a home-baked real bread type.

6. Discussion

To the best of the authors' knowledge, this is the first attempt in computer graphics to comprehensively model bread material using a physical model of bread baking. Previous work in bread material computation involved advanced tomography equipment [24] or artistic considerations [3]. These works are usually incomplete or inflexible to work with different bread types. To overcome these difficulties, our pipeline comprises different stages simulating bread making that allow a flexible and versatile modeling. This allows us to fine tune and test each step separately. Several bread types can be simulated by simply setting appropriate parameters for external shapes and the dough volume texture. This method allows us to approximate the appearance of any real bread crumb, by extracting parameters in order to accurately simulate it. The choice of geometry representation through volume textures provided a flexible approach that allows us to define the final shape, and to perform arbitrary cuts and slices in the geometries.

The proving step was inspired by visual similarities with Mandelbrot's model [16]. These patterns can be seen in several other porous materials such as foams, sponges, cheese, and many other. Users can define different parameters related to the amount of bubbles, their radii, and their mutual relationship, obtaining a flexible framework for porous geometries. The main visual difference between synthetic and real bread is that in the latter the bubbles in the crumb seldom intersect. Even though it is feasible to modify our dough generating algorithm to include a bubble self-avoiding feature, the computational cost would be too high.

A 1D baking model was employed, but generalized to fit 3D geometries using a discrete distance field, which retained the correctness and realism of the model. This choice does not seem to impair the quality of the results, reducing the model complexity and the computing resources consumed. The warping induced by the baking step shows acceptable results in the global and local deformations exerted to the dough volume texture. Also, since we physically emulated key steps in bread making, bubbles' shape adapted to the bread exterior silhouette, producing convincing patterns.

Bubble distributions under our model were shown to be statistically coincident to real bread, according to multifractal methods. Even though our bubble generation procedure initially employs mono-fractal generation, further steps in the modeling (like bubble intersection and deformation) generate structures that are not necessarily mono-fractal (i.e., the strict exponential scale invariance breaks). The sandbox multifractal analysis method produced very similar fractal features both with our synthetic images and with images of real bread crumbs. Two bread types were tested: a home-baked and the *baguette* bread types. The home-baked multifractal spectrum was more difficult to fit (i.e., the errors were larger than those of the *baguette* bread type), but notwithstanding this the final bubble distribution was adequate for emulating this bread type. Multifractal spectra showed higher dispersion in the negative dimensions ($q < 0$), which means that at finer geometric scales it is more difficult to characterize the statistical distribution, due to the image resolution. A more detailed link between bread crumb physical features (coarseness, porosity) and multifractal features may be useful for designing better procedural models [40].

Our pipeline can be regarded as among the first attempts to faithfully model bread making in computer graphics. The actual phenomenon is highly complex, not fully understood, involves human interaction, and includes several deformations and transformations of the dough material through processes that have been studied independently. To be computationally feasible, some of these processes require sensible simplifications, for instance the dough shaping by the cook, and some features arising in the crust, like cracking, or its actual color and texture. These simplifications are in line with the current state-of-the-art in understanding the bread making process, since in the specialized literature (for instance, in food engineering) there is no specific baking model that is both fully accepted and feasible to reproduce computationally. As in fluid dynamics simulations, to mention just an example, physically based modeling may lead to implementations that are computationally demanding and full of ad hoc considerations (convergence criteria, border conditions, and so on) which are not always adequate for a computer graphics application in which a more phenomenological solution would suffice. Further comprehension of the baking process, together with the increasing computational power, may hopefully lead in the near future to achieve a more physically sound modeling process than the one presented here.

While the pipeline might seem specialized only to model bread images, real pipelines in food production are similar for other materials (pizzas, cookies, croissants). If physical accuracy is not an issue, the modeling pipeline can be used directly, tuning parameters at different stages (colors, bubbling) in order to obtain credible images. In our pipeline, baking constitutes the bread-bounded step. If physical accuracy is desired, in order to be able to extend the

method for other materials, the baking stage should be replaced. The food engineering literature presents baking models for other foods [44,45] that can replace the bread baking model. In such a case, the validation section will remain the same and indeed it will be useful for designing and validating inner structures of those foods.

7. Conclusions and future work

In this paper we introduced a pipeline to model realistic arbitrary bread crumb and crust geometries based on a fractal generation procedure. The proving step is defined by the user in two ways: the bread global shape is determined feeding a 3D mesh model to the pipeline, and the bread local features are determined by setting different bubble distributions parameters (maximum and minimum bubble sizes, amount of bubbles, etc.). A physical simulation of the baking process produces bubble growth and coalescence, and also dough rise. Finally, the crust is generated around the exterior of the baked volume. The results were generated applying direct volume rendering, producing realistic images of crumb and crust for several bread types. This pipeline is far more flexible and powerful than other state-of-the-art approaches for modeling bread geometry. Multifractal image analysis was used to validate the model. The statistical similarity between 2D cuts in our results and of real-word bread slices suggests that it may be suitable for applications in 3D engines, serious games [46] and photorealistic rendering.

We are currently exploring several venues based on this work. One of the most useful features of our model is its ability to emulate different 3D dough textures. For this, we are aiming to fully automate the search in the parameter space, which will help to match several other bread types and materials. On the modeling side, we are considering inexpensive algorithms to introduce self-avoidance in the bubble generation model. Initial dough shaping will also be addressed. Crust generation is still ad hoc since a proper physical model is lacking, which gives space to artistic considerations (colors, knife cuts, cracks, etc.). Also, we plan to perform further tests on the image processing feature analysis, to gain insight in the multifractal bread characterization.

Acknowledgements

The authors wish to thank the anonymous reviewers, who helped to improve significantly the quality of this manuscript. R. Baravalle's work was supported by a doctoral scholarship of the National Science and Technology Council of Argentina (CONICET). This work was partially suported by research grant PGI 24/K060 of the SECyT-UNS, and the TIN2013-47137-C2-2-P project from Ministerio de Economía y Competitividad, Spain.

References

[1] Hughes JF, van Dam A, McGuire M, Sklar DF, Foley JD, Feiner SK, et al. Computer graphics: principles and practice (3rd ed.). Boston, MA, USA: Addison-Wesley Professional; 2013.
[2] Tong X, Wang J, Lin S, Guo B, Shum HY. Modeling and rendering of quasi-homogeneous materials. ACM Trans Graph 2005;24(3):1054–61.
[3] Xenakis A, Tomson E. Shading food: making it tasty for ratatouille. In: ACM SIGGRAPH 2007 courses, SIGGRAPH '07. New York, NY, USA: ACM; 2007. p. 22–33.
[4] JTK Jr, Raja S, Badler NI. Fruit senescence and decay simulation. Comput Graph Forum 2011;30(2):257–66.
[5] Liu Y, Yang X, Cao Y, Wang Z, Chen B, Zhang J, et al. Dehydration of core/shell fruits. Comput Graph 2015;47(0):68–77.
[6] Purlis E. Bread baking: technological considerations based on process modelling and simulation. J Food Eng 2011;103(1):92–102.
[7] Parish YIH, Müller P. Procedural modeling of cities. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. SIGGRAPH '01; New York, NY, USA: ACM; 2001, p. 301–8.
[8] Ebert DS, Musgrave FK, Peachey D, Perlin K, Worley S. Texturing and modeling: a procedural approach, 3rd ed. San Francisco, CA USA: Morgan Kaufmann Publishers Inc.; 2002.
[9] Müller P, Wonka P, Haegler S, Ulmer A, Van Gool L. Procedural modeling of buildings. ACM Trans Graph 2006;25(3):614–23.
[10] Prusinkiewicz P, Lindenmayer A. The algorithmic beauty of plants. New York, NY, USA: Springer-Verlag New York, Inc.; 1990.
[11] Rodriguez-Arenas O, Yang YH. Physically based baking animations with smoothed particle hydrodynamics. In: Proceedings of Graphics Interface 2011. GI'11; School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society; 2011. p. 17–24.
[12] Babin P, Valle GD, Chiron H, Cloetens P, Hoszowska J, Pernot P, et al. Fast x-ray tomography analysis of bubble growth and foam setting during breadmaking. J Cereal Sci 2006;43(3):393–7.
[13] Gonzales-Barron U, Butler F. Fractal texture analysis of bread crumb digital images. Eur Food Res Technol 2008;226:721–9.
[14] Mondal A, Datta A. Bread baking—a review. J Food Eng 2008;86(4):465–74.
[15] Prusinkiewicz P, Hammel M. A fractal model of mountains with rivers. In: Proceedings of graphics interface '93; 1993. p. 174–80.
[16] Mandelbrot BB. The fractal geometry of nature. 1 ed. New York: W.H. Freeman; 1982.
[17] Stam J. Stable fluids. In: Proceedings of the 26th annual conference on computer graphics and interactive techniques, SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.; 1999. p. 121–8.
[18] Fedkiw R, Stam J, Jensen HW. Visual simulation of smoke. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques, SIGGRAPH '01. New York, NY, USA: ACM; 2001. p. 15–22.
[19] Owens JD, Luebke D, Govindaraju N, Harris M, Krger J, Lefohn A, et al. A survey of general-purpose computation on graphics hardware. Comput Graph Forum 2007;26(1):80–113.
[20] Powathil G. A heat and mass transfer model for bread baking: an investigation using numerical schemes; 2004.
[21] Kruger J, Westermann R. Acceleration techniques for gpu-based volume rendering. In: Proceedings of the 14th IEEE visualization 2003 (VIS'03). VIS'03; Washington, DC, USA: IEEE Computer Society; 2003. p. 38.
[22] Nooruddin FS, Turk G. Simplification and repair of polygonal models using volumetric techniques. IEEE Trans Vis Comput Graph 2003;9(2):191–205.
[23] Osher S, Fedkiw R. Level set methods and dynamic implicit surfaces. New York: Springer Verlag; 2003.
[24] Van Dyck T, Verboven P, Herremans E, Defraeye T, Van Campenhout L, Wevers M, et al. Characterisation of structural patterns in bread as evaluated by x-ray computer tomography. J Food Eng 2014;123(0):67–77.
[25] Vanin F, Lucas T, Trystram G. Crust formation and its role during bread baking. Trends Food Sci Technol 2009;20(8):333–43. http://dx.doi.org/10.1016/j.tifs.2009.04.001.
[26] Zhang J, Datta A. Mathematical modeling of bread baking process. J Food Eng 2006;75(1):78–89.
[27] Thorvaldsson K, Janestad H. A model for simultaneous heat, water and vapour diffusion. J Food Eng 1999;40(3):167–72.
[28] Purlis Emmanuel, Salvadori Viviana O. Bread baking as a moving boundary problem. Part 2: model validation and numerical simulation. J Food Eng 2009;91:434–42.
[29] Gonzalez RC, Woods RE. Digital image processing. 3rd ed.. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.; 2006.
[30] Fan J, Mitchell J, Blanshard J. A model for the oven rise of dough during baking. J Food Eng 1999;41(2):69–77.
[31] Jefferson D, Lacey A, Sadd P. Crust density in bread baking: mathematical modelling and numerical solutions. Appl Math Model 2007;31(2):209–25. http://dx.doi.org/10.1016/j.apm.2005.08.017.
[32] Purlis E, Salvadori VO. Modelling the browning of bread during baking. Food Res Int 2009;42(7):865–70.
[33] Hunter RS. Photoelectric color difference meter. J Opt Soc Am 1958;48(12):985–93.
[34] Levoy M. Display of surfaces from volume data. Comput Graph Appl IEEE 1988;8(3):29–37.
[35] Max N. Optical models for direct volume rendering. IEEE Trans Vis Comput Graph 1995;1(2):99–108.
[36] Whitted T. An improved illumination model for shaded display. Commun ACM 1980;23(6):343–9.
[37] Singh J, Narayanan PJ. Real-time ray tracing of implicit surfaces on the gpu. IEEE Trans Vis Comput Graph 2010;16(2):261–72.
[38] Lafortune EP, Willems YD. Bi-directional path tracing. In: Proceedings of third international conference on computational graphics and visualization techniques (Compugraphics 93); 1993. p. 145–53.
[39] Jensen H. Global illumination using photon maps. In: Pueyo X, Schrder P, editors. Rendering techniques 96, Eurographics. Vienna: Springer; 1996. p. 21–30.
[40] Baravalle R, Delrieux C, Gómez JC. Bread crumb classification using fractal and multifractal features. Proceedings of the X Congreso Argentino de Mecnica Computacional (MECOM 2012); 2012. p. 3013–25.
[41] Tél T, Fülöp A, Vicsek T. Determination of fractal dimensions for geometrical multifractals. Physica A: Stat Mech Appl 1989;159(2):155–66.

[42] de Bartolo SG, Gaudio R, Gabriele S. Multifractal analysis of river networks: sandbox approach. Water Resourc Res 2004;40:2.
[43] Bosch M, Zhu F, Khanna N, Boushey Carol J, Delp EJ. Food texture descriptors based on fractal and local gradient information. In: European signal processing conference, EUSIPCO 2011; 2011. p. 764–8.
[44] Dumas C, Mittal GS. Heat and mass transfer properties of pizza during baking. Int J Food Prop 2002;5(1):161–77.
[45] Balaban M, Pigott G. Mathematical model of simultaneous heat and mass transfer in food with dimensional changes and variable transport parameters. J Food Sci 1988;53(3):935–9.
[46] Susi T, Johannesson M, Backlund P. Serious games: an overview. Technical Report, Institutionen för kommunikation och information, University of Skövde, Sweden; 2007.