ChatGPT

Procedural Rule Set for Transverse Dune Aeolian Systems

Hierarchical Multiscale Structure of Transverse Dune Fields

Transverse dune fields (ergs or sand seas) exhibit a distinct hierarchy of bedforms across multiple scales [1]. At the broadest scale, extensive **dune fields (ergs)** consist of many dunes arranged in **dune chains**. Often there are very large “mega-dunes” or **draas** (wavelengths of kilometers, heights of tens to hundreds of meters) with smaller dunes superimposed on their flanks [1]. Each individual **dune** is an elongated ridge oriented perpendicular to the prevailing wind, with a sharp **crest** line along its length. The upwind side (***stoss slope*) is a long gentle incline, while the downwind side forms a steep** lee slope **terminating in a** slip face **where sand avalanches occur** [2] **. Transverse dunes typically have a single slip face on the lee side and a lower-angle stoss side facing the wind** [2] **. The** interdune **areas between ridges are relatively flat corridors that separate adjacent dunes. Depending on sand supply, these interdunes may be sand-covered or exposed substrate: in sand-rich ergs, the interdunes are thin sand flats, whereas in sand-poor areas wind may scour the interdune down to hardpan or evaporite, creating a stark contrast (e.g. pale mud/salt flats between orange sand ridges)** [3] **. At the smallest scale, dune surfaces are covered with** wind ripples** (wavelength ~0.1–1 m) that run roughly perpendicular to the wind direction, adding fine texture to the stoss slopes [1]. This multiscale architecture – from regional dune field down to individual dune and ripple – must be captured to generate realistic analog images.

Physical Drivers and Controls

Several environmental factors control the formation and appearance of transverse dunes:

- **Wind Direction and Variability:** Transverse dunes require a **unidirectional wind regime** that blows consistently from one prevailing direction. A steady wind from one quadrant causes sand to accumulate in ridges oriented **transverse (perpendicular)** to the wind [4] [5]. Highly unidirectional winds (measured by a high RDP/DP ratio in drift potential analysis) favor continuous transverse ridges and barchans, whereas multidirectional or bimodal winds tend to produce other dune types (longitudinal or star dunes) [6]. For modeling purposes, one should assume a dominant wind vector and minimal seasonal change so that dune crest orientation remains uniform. If winds occasionally reverse direction (e.g. seasonally), dunes may develop minor reversing slip faces or even temporary flattening, but sustained bidirectional winds would ultimately create different morphologies (not pure transverse). Thus, in the transverse dune scenario the **wind rose** is strongly peaked in one direction.

- **Wind Strength (Shear Velocity):** The **velocity** of the wind controls sand transport capacity. Wind must exceed a threshold (the **entrainment threshold**) to move sand grains (~5–6 m/s at 10 m height for typical dry sand, corresponding to a threshold friction velocity u_{*t} of ~0.3 m/s). Above this, transport increases sharply – in fact, the sand flux rises roughly with the cube of wind speed beyond the threshold (Bagnold's law) [7]. Stronger winds move more sand and thus can build taller or more rapidly migrating dunes. In a generative model, this can be encoded in rules for sand transport: e.g. a **sand flux** q that is proportional to (u_*^3) or $(u - u_{\mathrm{thr}})^3$ (with u_* the friction velocity) [7]. High wind episodes can also **flatten smaller bedforms** – for example, a storm wind may temporarily erase ripples or cause sand to overshoot dunes – whereas moderate winds allow dunes to maintain their characteristic profiles. Over long periods, wind strength sets an equilibrium dune **size**: very fast winds tend to spread sand out (limiting vertical growth), while moderate winds allow higher dunes to persist. In modeling, one might set a maximum dune height tied to wind strength or atmospheric limits (e.g. dunes on Earth rarely exceed ~200–300 m because extremely tall dunes encounter wind shear patterns that curtail further growth [8]).

- **Sand Supply:** An ample **sand supply** is critical for transverse dune development. **Sediment availability** distinguishes barchan dunes from continuous transverse ridges. When sand supply is limited and ground surface has ample bare areas, dunes form as isolated barchans (crescentic mounds). With increasing sand supply, these individual dunes expand and coalesce into **barchanoid ridges**, and eventually a nearly continuous **transverse dune field** emerges once the surface is largely sand-covered [4]. In other words, as sand supply increases, isolated crescent dunes link up into longer ridges and the interdune areas fill in with sand [9]. For a procedural model, this means that the fraction of the surface covered by dunes versus exposed ground is a tunable parameter. A high sand cover (near 100%) produces uninterrupted transverse ridges, whereas a lower sand cover yields gap-toothed ridges or barchan chains. Sand supply also influences **dune size**: more sand generally allows larger dunes (until other limits like wind or area availability intervene). If the modeled environment has finite sand (e.g. a dune field bounded by non-sandy areas), dunes at the **upwind margin** might start as smaller barchans that grow and merge downwind as more sand accumulates.

- **Surface Moisture: Moisture** in the sand or at the surface inhibits sand transport by increasing inter-grain cohesion and raising the wind threshold for movement. Even in arid deserts, moisture can accumulate in interdune low points or beneath dune sands (e.g. near a water table or after infrequent rains). The presence of moisture tends to **stabilize dunes**: wet sand is less prone to erosion, so dunes in wetter climates (or seasons) migrate more slowly and develop smoother, fixed surfaces. For example, coastal transverse dunes often have a damp base or wet interdunal areas, which can lead to more pronounced interdune flats and even ponds. In semi-arid regions, precipitation can transiently bind the sand; studies show that water retained in dune soils is a major factor in stabilizing parabolic dunes [10]. In a generative model, one can incorporate moisture by specifying that certain zones (e.g. interdunes or lower elevations) have **reduced erodibility**. This would result in dunes with sharper boundaries (where dry, mobile sand meets a harder, moist surface) and potentially smaller dune size if moisture curtails sand availability. Over long timescales, alternating wet and dry conditions could freeze dune positions; however, in a static snapshot one might simply depict damp interdunes by using different color/texture and by showing more abrupt dune margins.

- **Vegetation: Vegetation** anchors sand and competes with dune-forming processes. In an active transverse dune field (erg), vegetation is typically sparse to absent due to intense aridity and moving sand. However, at the margins of dune fields or in coastal settings, plants can colonize dune surfaces and dramatically alter morphology. Vegetation **fixes sand in place**, encouraging development of *blowouts* and *parabolic dunes* (U-shaped dunes with trailing arms held by plants) instead of freemoving transverse ridges [11]. If the scenario includes vegetation, the rules would be: vegetation cover inhibits dune migration and favors a different dune shape (arms and lobes oriented upwind). For the purpose of modeling a *pure transverse dune system*, it's usually assumed vegetation is negligible. But one can still incorporate minor effects – e.g. a few stabilized patches in interdunes or along dune crests could create small notches or a slightly reduced slip face where vegetation holds sand. Generally, **no vegetation** is a good assumption for a large, active transverse sand sea, whereas **some vegetation** might be included to simulate semi-stabilized edges of a dune field (with corresponding changes in dune shape).

- **Grain Size and Sediment Characteristics:** The size and type of sand grains influence dune morphology. Transverse dunes typically consist of well-sorted fine to medium sand (≈0.1–0.5 mm). Finer grains are easily moved and can form sharper crests and well-defined slip faces. **Coarser grains** (or mixtures including granules) lead to dunes with lower relief and more gentle slopes because the heavier grains are harder for wind to lift. An extreme case is the formation of **zibar** dunes – low amplitude transverse ridges made of very coarse sand or granule-sized particles. In the Tenere Desert (Niger), for instance, large transverse ridges appear unusually symmetric and lack a pronounced slip face, which is attributed to their crests being armored with coarse grains (wind winnows away the finer fraction) [12] [13]. In modeling terms, one could adjust the **angle of repose** or the presence/absence of slipfaces based on grain size. Fine sand dunes would have the classic steep lee slope (~33°) and sharp crest, whereas a coarse-grained transverse ridge (zibar) might be modeled with a rounded, symmetric profile and no distinct slipface. Grain size also affects **color and reflectivity** in images (e.g. coarse sands may appear lighter due to a lag of quartz pebbles, etc.), which can be utilized when texturing the analog image. Finally, grain characteristics like the presence of dust or bimodal grain sizes can create streaks or patches (e.g. darker heavy mineral lag in interdunes) that can be added for realism.

- **Topographic and Boundary Constraints:** Real dune fields are sometimes influenced by underlying topography or boundaries (like a shoreline, a valley, or human obstacles). Generally, transverse dunes prefer relatively flat, open terrain – on significant slopes, sand will either accumulate at the slope base or form more complex shapes. If modeling a specific site, one must incorporate any large-scale slope (dune orientation may slightly bend if climbing a hill, and sand may thin on steeper terrain). Obstacles (rock outcrops, vegetation clumps) can create **shadow zones** of sand accumulation or cause local variations like spur dunes. In a generic model, one can assume an open flat area; however, if a **boundary** is present (for example, a non-sandy ridge or a mountain upwind that supplies sand), it could be represented by an upwind sand source feeding the dunes or by a zone where dunes terminate against a barrier. **Edge effects** include dune size tapering off at the field margins or dunes aligning along a boundary. Such effects should be included if relevant to the scenario (e.g. dunes ending at a desert–rock interface may form a line of barchans along the edge). Otherwise, one can impose periodic or open boundary conditions to simulate an extensive sand sea without artificial cut-offs.

By tuning these drivers – wind (directionality and strength), sand supply, moisture, vegetation, grain size – a generative model can capture a range of transverse dune environments. For a *true* transverse dune system (like an active central erg in a desert), assume: **strong, steady winds from one direction**, **abundant loose sand**, **dry conditions**, **no vegetation**, and **fine sand**. These assumptions will yield the classic transverse dune field pattern.

Time-Evolution Rules and Dune Dynamics

To ensure geological accuracy, the model should incorporate rules that emulate how transverse dunes develop and change over time. Key processes include dune initiation, migration, interaction (merging and splitting), and smaller-scale bedform dynamics:

- **Dune Formation and Growth:** Transverse dunes begin as smaller sand waves or protodunes that emerge from a flat sand bed under wind action. Small perturbations on the sand surface grow at a **predictable initial wavelength** (the most unstable wavelength for the given wind and sand conditions) [14]. On Earth, initial dune spacings are on the order of tens of meters. These nascent dunes increase in amplitude via positive feedback: once a mound forms, it slows the wind slightly, causing more sand to drop out and the mound to grow. Over time, continuous winds drive these baby dunes to coalesce into larger forms. In a simulation, one could start with a noise or a spectrum of small ridges and allow the dominant wavelength to amplify. **Dune growth** in height is ultimately limited by wind flow adjustments – as dunes get taller, airflow acceleration over the crest causes sand transport to bypass the dune (or avalanches to occur) rather than indefinitely building up. One elegant hypothesis is that dune heights are capped by the altitude of the atmospheric boundary-layer (mixed layer) – once a dune approaches a size that disturbs the entire lower atmosphere, it cannot grow taller [8]. This implies a natural maximum height in the model (for Earth ~200 m, as noted). Practically, time-evolution rules should allow dunes to grow when they are below equilibrium size (adding sand to their volume) and stop growing or spread laterally once they reach a certain height aspect ratio.

- **Dune Migration:** Under a steady wind, transverse dunes **propagate downwind**. The mechanism of movement is through erosion and deposition of sand: wind impacts the stoss side, picking up grains (via creep and saltation) and carrying them up the slope. Upon reaching the crest, the sand grains fall over to the lee side and accumulate until the **slip face** periodically avalanches. Each avalanche (a small landslide down the lee) adds a layer of sand to the dune's downwind side and effectively moves the dune forward by a short distance. Internally, this creates diagonal **cross-bedding** that is inclined in the direction of migration (a hallmark of aeolian sandstone stratification) [15]. In modeling, the rule is: *erode from the upwind side, deposit to the downwind side*. The **rate** of dune migration depends on sand flux and dune size. A simple quantitative rule is that the migration velocity v is inversely proportional to dune height. Essentially, a given sand transport rate will move a low dune more quickly than a tall dune (since the tall dune has more volume to shift). This can be expressed as:

$$v \approx \frac{Q}{H},$$

  where Q is the volumetric sand flux per unit width and H is dune height [16]. In other words, dunes move by the amount of sand that can be delivered over their height [16]. In real deserts, small dunes (a few meters high) can travel astonishing distances – for example, dunes ~3–5 m tall have been observed to migrate over **100 m per year** under strong winds [17]. In contrast, large dunes (tens of meters high) might crawl only a few meters per year or even less. A time-evolution rule should reflect this: for instance, compute sand flux from wind speed (e.g. using Bagnold's formula) and then update each dune's position by an amount $\Delta x = (Q/H)\Delta t$ each timestep. If implementing on a grid, one could move sand cells downwind proportional to local flux and inversely to local elevation.

The model should also maintain the **angle of repose** on the lee side: whenever sand accumulates such that the lee slope exceeds ~33°, an avalanche is triggered moving that sand one cell further downwind (this keeps the profile realistic and is analogous to real slip-face dynamics). This avalanche rule effectively *locks* the slope at the critical angle, producing the proper steep lee in the static snapshots.

- **Crestline Interactions and Merging:** In a field of multiple transverse dunes, dunes will interact as they migrate. **Crestline interactions** include merging of dunes, splitting (calving) of dunes, and lateral linking of adjacent segments. Over time, there is a tendency toward **pattern coarsening** – small dunes merge into larger dunes, increasing the average wavelength of the field [18]. The model should allow for such interactions. For example, if a faster-moving smaller dune approaches the back of a slower larger dune, one possible outcome is it will merge – effectively adding its sand to the bigger dune and increasing that dune's height or length. This can be handled by a rule: *if a migrating dune catches up to another, combine them into a single dune with volume equal to sum of both*. Another observed outcome is **dune collision and off-spring formation**. Sometimes when a smaller dune hits a larger one, it doesn't fully amalgamate but instead causes a portion of the large dune's sand to spill forward, creating a new dune in front – this is dune **calving** (analogous to how a breaking wave might spawn a new pulse). The rules can accommodate this by: *if a small dune overtakes a big dune above a certain relative size threshold, spawn a new dune downwind (of intermediate size) and reduce the big dune accordingly*. Empirical studies and simulations have shown dunes can behave like solitons, exchanging mass on collision – but for simplicity, one can choose either merge or calve as deterministic outcomes based on size ratio or on randomness [19]. Additionally, **crestline elongation** and lateral linking occur: transverse dunes may extend their ends (flanking barchan horns) and link with neighboring dunes along the sides. The model could check gaps between crest segments – if sand is plentiful and wind slightly off-normal, sand can accumulate in the gap, fusing two segments into one longer ridge. Over time, this removes defects (gaps) from the field, increasing transverse continuity. Including a rule for lateral growth (e.g. dunes extending if there is a nearby adjacent dune within some distance) will help simulate the natural trend toward more continuous crestlines over long times.

- **Dune Splitting and New Dune Formation:** The opposite of merging is **dune splitting**. Transverse dunes can generate new dunes through a couple of mechanisms. One is the aforementioned calving, where a slipface collapses forward into a new dune. Another is if wind dynamics create an internal **wavelength doubling** – sometimes a very large transverse ridge might develop an extra crest in its lee (especially if wind strength increases or direction shifts slightly). This can be thought of as the big dune shedding its excess into smaller forms. In cellular automata models, this is often represented by a dune reaching a critical size and then emitting a "daughter" dune downwind. To incorporate splitting, one can set a rule: *if dune height exceeds H_max or if crest length exceeds a certain multiple of its wavelength, form a new small dune on the downwind side*. This keeps the size distribution in check and prevents unrealistic single dunes from growing indefinitely. In real ergs, such splitting helps maintain a field of dunes rather than one giant dune.

- **Ripple Migration and Surface Bedforms:** On each dune's surface, smaller **bedforms (ripples)** are constantly forming and migrating. These do not substantially alter the dune's overall mass or position but are an important visual and sediment-transport feature. Wind ripples move *much* faster than dunes – they can migrate on the order of centimeters to meters per day. In the model, one doesn't need to track individual ripples in detail (that would be too fine-scale), but their effect can be included. For instance, one could superimpose a high-frequency oscillation on the sand transport to mimic ripple translation. A simpler approach for static images is to **add ripples as texture** (small ridges on stoss slopes). However, for completeness: ripple formation rules could state that *any flat sand surface under wind will develop ripples of wavelength ~0.2–1 m oriented perpendicular to wind*. If simulating time evolution, one might periodically re-impose a ripple pattern on the dune surfaces and let it ride downwind. The presence of ripples slightly increases the effective roughness of the dune surface, which can slow the wind right at the surface and slightly modulate how sand is picked up (this is a secondary effect often ignored in large-scale models, but could be implicitly included by adjusting the sand transport coefficient on rippled surfaces). In the context of the analog image, ensuring that dunes are not perfectly smooth is important – the lee slope may appear smooth (sand avalanches sort it into a clean slope), but the windward slope should show subtle ridges (ripples) aligned horizontally.

- **Time-Scale and Stratigraphic Evolution:** While the primary goal is generating a surface image, it's worth noting the long-term stratigraphic record that transverse dunes create. As dunes migrate, they leave behind sets of cross-bedded layers inclined downwind. Interdune areas, if occasionally wet or vegetated, can leave flat-lying deposits (e.g. ephemeral lake or soil deposits) that appear as horizontal strata between dune cross-bed sets [20]. If the analog modeling extends to 3D or cross-sectional views, rules for **stratal geometry** can be added: e.g. when a dune moves a certain distance, drop a foreset package representing that motion. For the surface model, we mainly ensure that the **current morphology** is consistent with having formed by these processes. For example, large dunes should have proportionally wider spacing (because of past merging/coarsening), slipfaces should all face downwind (due to consistent winds), and any irregularities in crest shape should be explicable by recent interactions (e.g. a notch in a dune might indicate where two dunes merged or a gust cut through). The **age** of the dune field could be inferred by how organized it is – a very young field might have many small dunes and chaotic arrangement, whereas a mature field shows regular, aligned ridges. Thus, depending on the desired level of organization in the image, one can qualitatively dial the simulation time. Running the evolution rules longer will result in larger, more widely spaced, and more uniform transverse dunes (reaching a quasi-equilibrium pattern), whereas a short simulation will produce a more irregular, hummocky pattern of smaller dunes. In summary, the time-evolution rules (migration, merging, splitting, etc.) collectively drive the model toward a realistic steady-state configuration of transverse dunes, and they ensure that any snapshot taken from this model "history" will look geologically plausible.

Static Generation Rules for a Transverse Dune Field (Snapshot Construction)

To create a geologically plausible **snapshot** of a transverse dune field, we apply the learned principles in a procedural way. The goal is to generate a static 2D (or 3D) representation that *looks* like a real transverse dune landscape, respecting the spatial patterns and relationships dictated by physics. Key guidelines:

1. **Orientation and Alignment:** Choose a consistent prevailing wind direction for the scene (e.g. from east to west). All dunes should be oriented **perpendicular** to this wind. This means drawing each dune ridge roughly linear and oriented along, say, a north–south axis if wind is east–west. In practice, one can set a global wind angle θ and then orient every dune's crest normal to θ. Ensure **lee**

slopes are on the downwind side uniformly. For instance, if wind blows westward, every dune's slipface must face west. This avoids any contradictory dune orientations in the snapshot.

2. **Dune Spacing and Size:** Impose a characteristic **wavelength (λ)** and **height (H)** for the dunes, based on known scaling relationships. Field measurements show that transverse dunes often have crest-to-crest spacing on the order of a few tens to a few hundreds of meters [1]. As a rule of thumb, the spacing might be about 5–15 times the dune height, depending on sand supply and wind steadiness. For example, a 10 m tall dune might be separated from the next dune by ~100 m of interdune trough. In the model, pick an average dune height (e.g. H = 15 m) and use a proportional spacing (e.g. λ = 10 H = 150 m). Introduce some randomness (±20–30%) in spacing and height to mimic natural variability. The **distribution** of dune heights can be normal or log-normal around the mean H. No two dunes should be exactly identical in size – real dunes show a spectrum of sizes, though in a mature transverse field the sizes cluster within the same order of magnitude. If the field is extensive, you might also incorporate subtle large-scale variation, such as slightly taller dunes in the center of the erg where sand may accumulate more, and slightly smaller toward the edges.

3. **Crest Sinuosity and Continuity:** Perfectly straight dune crests are rare; instead, transverse dune crests often exhibit gentle undulations or limited breaks (defects). To recreate this, generate each dune's **crestline** with a slight sinuous curvature. For instance, one can perturb a straight line by a low-amplitude sine wave (wavelength along-crest perhaps a few times the dune spacing). The amplitude of crest sinuosity should be a small fraction of the spacing (e.g. crest deviates left-right by at most 10% of its length). Also decide if the crest is **continuous across the entire model domain** or if it has gaps. In areas of high sand supply, transverse ridges can run unbroken for many kilometers. In other cases, the ridges are composed of serial barchanoid segments with junctions. You can emulate segmented ridges by introducing occasional small gaps or offsets in the crestline – effectively splitting one ridge into two aligned dune segments with a short gap. Any gap should be on the order of a dune spacing or less, and the offset segments should overlap slightly so the overall ridge trend is maintained. Where a gap exists, the ends of the dune segments might be curved like the horns of a barchan. This adds realism by showing some dunes still merging. Ensure that **neighboring crestlines alternate** properly: you wouldn't want gaps in adjacent dunes to all line up – in reality they tend to be staggered, giving a checkerboard pattern of defects. This can be achieved by randomizing which dunes have a gap and where, with some exclusion rules to prevent vertical alignment of gaps.

4. **Dune Profile Shape:** Assign each dune a cross-sectional **profile** that is consistent with aeolian shapes. A convenient idealization is to use an asymmetric **sine wave** or **Gaussian** for the dune cross-section. For example, for a given dune height H and half-wavelength (from crest to trough) *L_stoss* upwind and *L_lee* downwind, you can define the height profile along a wind-parallel slice through the dune as:

5. *Stoss side (upwind of crest):* $z(x) = H \exp\left(-\frac{x_{\text{crest}}-x}{L_{\text{stoss}}}\right)$ for $x_{\text{upwind}} < x < x_{\text{crest}}$. This creates a gentle exponential ramp up to the crest. Alternatively, a linear ramp or convex-up curve can be used (many dunes have approximately planar stoss slopes with slight upward curvature near the crest).
6. *Lee side (downwind of crest):* $z(x) = H\left(1 - \frac{x-x_{\text{crest}}}{L_{\text{lee}}}\right)$ for $x_{\text{crest}} < x < x_{\text{toe}}$, with a cutoff at zero height. This produces a straight linear slipface at ~33° if L_{lee} is chosen accordingly. For instance, if the dune height is 15 m, setting $L_{\text{lee}} \approx 25$ m yields a slope ≈ 15/25 = 0.6 (≈31°). You can refine this

by making the upper part of the lee slightly concave to simulate initial avalanche curvature, but a linear slope is a good approximation of an avalanche face.

7. Beyond the dune's "toe" (the downwind foot), the height returns to the base level (interdune elevation). The interdune may even dip below the base if deflation has occurred (could be modeled as a slight negative bump).

Ensure that the **maximum slope** of the lee does not exceed the angle of repose (~33°) in the final discretization. If using a grid, you will need to enforce a slope limit: any cell that has a drop greater than ~0.6 in height ratio to its neighbor should be adjusted (avalanched) until the slope is at or below that value. The stoss slope in contrast is much gentler (often 5°–15°). You might simply choose L_{stoss} such that the stoss angle at the crest is ~10°. Note that dunes can have a concave stoss near the base (due to scour in the interdune) and a convex rollover at the crest – for simplicity a single smooth function is fine, or you can piecewise define a concave-up lower stoss and concave-down near-crest. In summary, the dune's 2D section should look like a skewed hill: long shallow ramp and abrupt steep drop.

1. **Interdune Surface:** Determine the **base elevation** or reference ground level. Often in models we take base = 0 and build dunes upward. However, real interdunes may not all be at the exact same elevation – some interdune areas are slightly lower due to wind scour or slightly higher due to underlying topography or vegetation trapping sand. For a realistic snapshot, introducing minor elevation variability in the interdune flats helps. For instance, you could allow interdune elevation to vary by a meter or two over the field (with maybe a very low-frequency undulation or a gradient). If there are known interdune features (like dry lake beds or vegetation patches), those can be represented by localized higher or lower zones. On the image, **interdunes should appear smoother and flatter** than dunes, possibly lighter in color if they are hardpan or salt-encrusted flats [12]. Model-wise, after drawing dunes, you can **subtract a small height** uniformly from all non-dune areas to simulate slight deflation in interdune areas (making them lower than an arbitrary datum). It's also important to ensure continuity: the transition from dune toe to interdune should be smooth (unless you specifically want a sharp "cliff" which is unusual in sand – dunes normally grade into the interdune surface). One might apply a slight Gaussian blur or smoothing to the base of dunes so they blend naturally into the flats.

2. **Sand Volume Conservation:** While constructing dunes, keep an eye on overall sand volume if the scenario calls for it (especially if simulating source vs sink areas). In a strictly static image one might not worry, but it adds realism if, for example, areas immediately downwind of a gap have a little sand shadow (less sand because it's in the dune). A procedural rule: *do not arbitrarily pile sand without accounting for source.* If an area is entirely within an erg, we assume plenty of sand. But if you have a fixed sand budget, you might distribute a certain volume of sand per unit area in dune form. This ensures the interdune exposure corresponds to how much sand is tied up in dunes. For practical generation, this means not making dunes too tall or too closely spaced beyond what the sand amount would allow. An approximate check: the cross-sectional area of all dunes in a field slice should equal the total sand thickness that would exist if the area was flat and covered with sand. If dunes are drawn too big and too tightly packed, the model would implicitly be creating excess sand out of nowhere. So a more advanced static rule is: adjust dune heights or spacing until the total sand volume matches an expected value (e.g. if sand cover is 80% with average depth 5 m, then distribute that into dunes accordingly).

3. **Superimposed Dunes and Ripples:** For added realism, especially in large dune fields, consider **multi-scale superposition**. On very large transverse ridges (draas), smaller dunes often exist on their backs. For example, a mega-dune hundreds of meters high might have ordinary dunes (tens of meters high) on its flanks, and those dunes in turn have ripples. If you want to simulate a complex dune field, you can do this hierarchically: first generate a broad undulating surface for mega-dunes, then on top of that surface, apply the transverse dune algorithm for the smaller dunes. This will yield a **compound dune** appearance. Each scale should use its appropriate spacing and size (mega-dune spacing kilometers, dunes tens to hundreds of meters, ripples sub-meter). If doing this, ensure that the smaller dunes are oriented and migrate in concordance with the big dunes (usually the same wind, though sometimes mega-dunes can have slightly different orientation if ancient wind regimes differed). For simplicity, a single-scale model (one size of dune) is often sufficient for a snapshot unless the scene explicitly warrants mega-dunes. At minimum, do include **wind ripples** as texture on the stoss slopes. This can be done by overlaying a fine undulating pattern (perlin noise or sinusoidal stripes) oriented perpendicular to wind on each dune's windward side. The ripple spacing can be, say, 0.5 m and amplitude a few centimeters. Visually, this detail signals active wind processes at a smaller scale.

4. **Shadows and Light Orientation:** When rendering the analog image, keep in mind the sun angle because dune topography produces distinctive shadow patterns that reveal shape. Typically, satellite images are taken with sun elevation 30–60° and from some angle. To illustrate dunes, it's best if the sun is not directly overhead – low-angle light casting shadows of the crest onto the interdune enhances the ridges. In the generation framework, after shaping the dunes, you can simulate lighting to verify the appearance. **Slopes facing away from the sun should appear in shadow** and slopes facing the sun bright. This will accentuate the slipfaces if lit from a low angle on one side. For instance, in a snapshot with sunlight from the northeast, the southwest sides of dunes (assuming wind from east) will be illuminated and the northeast sides (slip faces if wind is from east) will be in shadow – opposite lighting would highlight the slip faces instead. Choose a lighting that matches the typical look of desert dunes (often images are taken with sun from the southeast or northwest to cast long shadows). This is not a generation "rule" per se, but it is crucial for *visual accuracy*. In procedural terms: compute illumination = max(0, dot(normal, sun_direction)), and use that to shade the heightfield.

By following these static generation rules, one constructs a snapshot where dunes have the correct orientation, spacing, shape, and surface patterns. For example, you will end up with a series of subparallel ridges, each ridge showing a long, shallow slope facing the wind and a sharp slipface on the lee. The spacing between ridges will be regular but not perfectly so, and the troughs will align somewhat continuously across the field. The image should resemble an authentic aerial photo of a transverse dune erg, with no obvious tiling artifacts or unrealistically geometric features. The next section provides an example and visual references to guide this layout.

Visual Example and Spatial Layout Guidance

Example of a transverse dune field (satellite/astronaut photograph from the Tenere Desert, Niger). Long, dark-toned transverse dunes run roughly north–south, perpendicular to the dominant northeasterly wind [21]. The dunes have linear to gently wavy crests and consistent spacing. Sunlight from the east highlights the steep slip faces on each dune's western side by shadow (dark side of each ridge), while the interdune flats appear as broad, lighter-toned areas between dunes [12]. This natural example illustrates key layout characteristics: uniform *orientation of dunes, parallel ridge pattern, and alternating bands of high sand accumulation (dune crests) and low areas (interdunes).* [12]

When generating an analog image, it is helpful to mimic such real patterns. From the figure, note how the transverse ridges are fairly continuous, and smaller-scale features ride on the larger forms. The crests are not perfectly straight – they meander slightly and sometimes fork. Interdune areas in this image are flat and somewhat wider than the dune crests, indicating abundant but not complete sand cover. In some other satellite images (e.g. the Grand Erg Oriental in Algeria), transverse dunes appear as sharper ridges with narrower interdunes, or as barchanoid chains with more pronounced scalloping of the crests. One should decide the specific style (continuous vs. barchanoid transverse dunes) for the model and use real photos as templates. Schematic diagrams in textbooks (which show cross-sections with slipfaces, or plan-view sketches of barchan-to-transverse transitions) can also guide proportions. For instance, diagrams typically illustrate a ~120° angle between the horns of barchans, which, when merged, become the undulating crest of a transverse ridge. Translating that to the model: the crest undulations can have a curvature similar to barchan horns to look natural.

If available, **satellite DEMs or topographic profiles** of known dune fields can be used to calibrate the model. These provide quantitative measures of crest spacing variability, dune height distribution, and the shape of dune cross-sections. For example, LIDAR scans of White Sands (a smaller dune field) show nearly sinusoidal transverse dune profiles with slipface angles ~32° and an average spacing around 100 m for 10 m high dunes. One could directly use such a profile curve in the generation rules. Likewise, high-resolution images from Google Earth or NASA imagery can be sampled to extract ripple spacing and defect frequency. Using these as references ensures the synthetic image falls within realistic bounds (e.g. not too orderly or too random).

In summary, grounding the model output in real-world examples is crucial. The embedded image above serves as a visual benchmark: the synthetic dunes should have a comparable appearance – aligned ridges with coherent slipfaces and natural variation. The **spatial layout** (the planform pattern) is essentially a rhythmic repetition of dunes, and by inspecting real patterns one can adjust the model's dune spacing and alignment until it "looks right." The goal is that someone familiar with aeolian geology would instantly recognize the output image as a transverse dune depositional system, and not, say, linear dunes or a fluvial pattern. Matching the key visual cues (parallel curvy ridges, lee-side shadows, consistent orientation) will achieve that recognition.

Pseudocode for Generating a Transverse Dune Field Model

Finally, we outline a high-level pseudocode framework that could be implemented to generate the described dune field in a simulation or graphics program. This pseudocode integrates the rules above into an algorithmic sequence:

```
# Define model parameters
wind_direction = θ                        # e.g., θ = 90° for wind from east to west
domain_size_x, domain_size_y = Lx, Ly     # domain dimensions (Lx along wind, Ly across wind)
grid = initialize_heightfield(Lx, Ly)     # 2D array for elevations, start at base
```

```
level (0)
dune_height_mean = H0                   # e.g., H0 = 15 m
dune_spacing_mean = λ0                  # e.g., λ0 = 10 * H0 = 150 m
dune_height_variation = 0.3             # 30% random variation
dune_spacing_variation = 0.2            # 20% variation in spacing
angle_of_repose = 33°                   # critical slope angle

# Calculate number of dune crests to generate based on domain length and spacing
N_dunes = ceil(domain_size_x / dune_spacing_mean) + 2   # +2 to cover edges

# Loop to place each transverse dune ridge
for i in range(0, N_dunes):
    # Determine the downwind position (x-coordinate) of this dune’s crest
    base_x = i * dune_spacing_mean
    x_crest = base_x + rand_uniform(-dune_spacing_variation,
dune_spacing_variation) * dune_spacing_mean

    # Skip if crest is outside domain (for edges)
    if x_crest < 0 or x_crest > domain_size_x:
        continue

    # Assign a height for this dune
    H = dune_height_mean * (1 + rand_normal(0, dune_height_variation))
    H = max(H, 0)   # no negative heights

    # Optional: apply a large-scale undulation for mega-dune (not shown for
simplicity)

    # Generate crest line across y-direction (width of domain)
    # We represent the crest as a centerline along x = x_crest, with small
lateral sinuous perturbation.
    crest_curve = []
    phi = rand_uniform(0, 2π)   # random phase for sinuosity
    crest_amplitude = 0.1 * dune_spacing_mean   # e.g., 10% of spacing
    crest_wavelength = 2 * dune_spacing_mean    # e.g., crest undulation
wavelength
    for y in range(0, domain_size_y):
        # small sinusoidal deviation in x position for the crest
        delta_x = crest_amplitude * sin(2π * y / crest_wavelength + phi)
        crest_curve.append(x_crest + delta_x)
    # The crest_curve now holds the x-coordinate of the crest for each y across
the domain.

    # Raise the dune shape centered on crest_curve.
    for y in range(0, domain_size_y):
        x_center = crest_curve[y]
        # Iterate along wind (x direction) around the crest to build the dune
        # We'll add height contributions from this dune to the grid.
```

```
        # Define characteristic lengths for stoss and lee based on height
        L_stoss = 8 * H    # (example: stoss length ~8x height)
        L_lee   = 4 * H    # (example: lee length ~4x height, gives ~14° slope
if 4x, but we will adjust below)

        # Upwind side: from x_center - L_stoss to x_center (crest)
        for x = floor(x_center - L_stoss) to floor(x_center):
            if x < 0 or x > domain_size_x:
                continue
            # Compute height profile on stoss (e.g., exponential or linear ramp)
            dist_upwind = x_center - x    # distance from this point to crest
            z = H * exp(-dist_upwind / L_stoss)    # exponential decay upwind
            # Accumulate this dune’s contribution on the grid (taking care of
multiple dunes)
            grid[x, y] = max(grid[x, y], z)        # use max to avoid
overlapping dunes summing unnaturally

        # Downwind side: from x_center to x_center + L_lee
        for x = floor(x_center) to floor(x_center + L_lee):
            if x < 0 or x > domain_size_x:
                continue
            dist_downwind = x - x_center   # distance from crest
            # Linear decrease downwind until reaches 0 at toe (x_center + L_lee)
            z = H * (1 - dist_downwind / L_lee)
            if z < 0:
                z = 0
            grid[x, y] = max(grid[x, y], z)
```

```
    # End of loop over domain width (y) for current dune
endfor   # next dune ridge

# After initial placement of all dunes, apply post-processing to enforce
realistic slopes and cleanup

for each cell (i, j) in grid:
    # Enforce angle of repose (simple approach):
    # Check neighbor in downwind direction (i+1, j since wind from left to right
in this setup)
    if i+1 < domain_size_x:
        height_diff = grid[i, j] - grid[i+1, j]
        slope = atan(height_diff / grid_spacing)   # grid_spacing = cell size
        if slope > angle_of_repose:
            # Move excess sand downwind
            excess = (tan(slope) - tan(angle_of_repose)) * grid_spacing
            grid[i, j] -= excess
            grid[i+1, j] += excess
```

```
    # (In practice, one would need to iteratively relax slopes; this is a one-pass example)

# (Optional) Smooth interdunes slightly and adjust for any negative values
apply_small_smoothing_filter(grid, only_in_interdune_regions)

# (Optional) Add ripple texture:
for each cell (i, j) in grid:
    if grid[i, j] above some threshold (i.e., it's on a dune stoss slope):
        ripple_amp = 0.05  # 5 cm
        ripple_wavelength = 1.0  # 1 m
        # Add a sinusoidal perturbation based on position along wind:
        grid[i, j] += ripple_amp * sin(2π * i / ripple_wavelength)
```

```
# The grid now contains a heightfield of transverse dunes. Next steps would be rendering:
render_heightfield(grid, sun_angle, camera_angle, etc.)
```

In this pseudocode, we:

- Initialize a flat grid and determine how many dunes (crests) will be placed based on the domain length and desired spacing.
- Loop over each dune crest index `i`, computing a perturbed `x_crest` position and a random height `H` for that dune.
- Create a **crest curve** across the domain width with a slight sinusoidal variation. This models a wavy crest line instead of a perfectly straight one.
- For each point along that crest (each y), we “raise” a dune shape. We use an exponential decay for the stoss side and a linear drop for the lee side as a simple shape function. We add height to the grid, taking the maximum to avoid overlapping dunes stacking on each other (this assumes dunes farther upwind or downwind won’t overlap much; if they do, one might sum and then clip, but max tends to work for distinct ridges).
- After placing all dunes, we enforce the **angle of repose** constraint by iterating through the grid and checking slopes. Here a simple check looks at each cell and its immediate downwind neighbor; if the slope exceeds the repose angle, we move excess sand down (this mimics avalanching). A more sophisticated approach might check diagonal neighbors too (since avalanches can spread sand laterally a bit) and iterate until no cell exceeds the threshold.
- We then smooth the interdune areas to avoid any blocky artifacts. For example, one can run a mild smoothing filter but mask out the dune crest areas to preserve sharpness.
- Finally, we superimpose a ripple pattern: in the pseudocode, we added a sinusoidal height perturbation with a 1 m wavelength and a few centimeters amplitude on the stoss sides (you’d actually want to distinguish stoss vs lee; here we might apply it everywhere on the dune except maybe skip the slipface cells).
- The resulting heightfield can be rendered or output. If rendering, apply a sand color, and a light source. The shadows and highlights will automatically appear from the geometry.

This pseudocode is just one approach – alternative implementations could use cellular automata for the evolution or spectral methods to impose a repeating pattern. The key is that the rules enforced (orientation, spacing, slipface slopes, etc.) ensure the output is a **physically and geologically consistent** transverse dune system.

When this code (or similar algorithm) is executed, the expected outcome is an image or model of dunes that captures the essence of Figure 1 above: a series of transverse ridges of sand, with realistic spacing, height, and morphology, as would be found in a natural aeolian depositional environment [12]. Each ridge in the model will have a gentle windward slope and a steep lee slope, and the overall pattern will reflect the prevailing wind and ample sand supply that characterizes transverse dune fields. By adjusting parameters (wind direction, sand amount, dune size), the framework can simulate different transverse dune scenarios (e.g. tighter barchanoid ridges vs. very large wide-spaced dunes), all within the envelope of geologic plausibility.

---

[1] [6] (PDF) Dune Morphology and Dynamics
https://www.researchgate.net/publication/237080611_Dune_Morphology_and_Dynamics

[2] [4] [5] [9] [15] [17] Dune - Wikipedia
https://en.wikipedia.org/wiki/Dune

[3] Dune Types in the Issaouane Erg, Eastern Algeria
https://earthobservatory.nasa.gov/images/6910/dune-types-in-the-issaouane-erg-eastern-algeria

[7] Bagnold formula - Wikipedia
https://en.wikipedia.org/wiki/Bagnold_formula

[8] [14] [18] [19] What sets aeolian dune height? | Nature Communications
https://www.nature.com/articles/s41467-022-30031-1?error=cookies_not_supported&code=d3c0bd6d-7fb3-4929-8312-b60fbf30483e

[10] [11] 19.3: Dunes - Geosciences LibreTexts
https://geo.libretexts.org/Courses/Lumen_Learning/Earth_Science_(Lumen)/19%3A_Deserts/19.03%3A_Dunes

[12] [13] [21] Sand Dunes in the Tenere Desert, Niger
https://earthobservatory.nasa.gov/images/42070/sand-dunes-in-the-tenere-desert-niger

[16] Prediction of barchan dunes migration using climatic models and ...
https://www.sciencedirect.com/science/article/pii/S0012821X24004813

[20] 5 Outcrop photograph of the Navajo Sandstone, a Jurassic aeolian...
https://www.researchgate.net/figure/Outcrop-photograph-of-the-Navajo-Sandstone-a-Jurassic-aeolian-sandstone-at-Zion-Canyon_fig4_278716791