IEDCO Parallax

Visual depth tuning system for React and Angular interfaces

EDITION

Revised edition
October 22, 2026

AUTHOR

Rob Vance

PRODUCT

IEDCO Parallax for React and Angular (beta)

Depth as a system

Coordinated scroll depth for React and Angular interfaces

EDITION	Revised edition • October 22, 2026
AUTHOR	Rob Vance
SOURCE MANUSCRIPT	Based on IEDCO Parallax: Coordinated depth and scroll behavior for React and Angular (draft, October 22, 2026).
PRODUCT	IEDCO Parallax for React 1.1.0 and for Angular 1.1.0 (beta), an add-on for IEDCO Motion
SOURCES CHECKED	Eighteen references, each tested and read on October 22, 2026 (Appendix B)

> **Attribution note:** IEDCO Labs publishes both IEDCO Parallax and this paper, so the paper is not an independent evaluation. Product facts come from the product's own documentation and images. The cited sources support the general mechanics and requirements only; none of them tests IEDCO Parallax, and the product's footprint, performance, and comfort claims remain hypotheses until measured.

ABSTRACT

Parallax creates apparent depth by moving visual layers at different rates, and on the web its usefulness depends on a clear relationship between user input, content position, and visual movement. This paper examines scroll-driven depth as part of an interface's motion system rather than as an isolated effect, using IEDCO Parallax, an add-on to IEDCO Motion for React and Angular, as the case study. It draws on perception research on motion parallax, the W3C specification for scroll-driven animation, browser rendering guidance, framework lifecycle documentation, and accessibility guidance that names parallax directly. The resulting approach distinguishes scroll progress from elapsed time, orders measurement before style changes, keeps continuous updates outside component rendering, shares timing and reduced-motion settings with the surrounding motion system, and removes decorative depth when motion is reduced. The central argument is that controlled depth should belong to the rest of the interface, and that claims of smaller footprint, better performance, or greater comfort require scene-level measurement. Priorities for later versions and the evidence each claim still needs close the paper.

Keywords: parallax; scroll-driven animation; motion parallax; depth cues; rendering performance; React; Angular; reduced motion; accessibility

1.

Introduction: depth effects built in isolation

Parallax is easy to demonstrate and harder to integrate well. A background moves more slowly than the foreground, and the page acquires an immediate impression of depth. The difficulty appears when that effect joins a real page containing responsive layouts, variable content, navigation, interactive controls, and other animation.

Independent effects can develop conflicting behavior. A background drifts slowly while cards enter sharply. A section continues moving after the user expects it to settle. Cropping on mobile screens exposes empty space. Content changes invalidate the measurements used to calculate movement. These problems share a cause: the effect has been treated as a visual addition rather than as part of the application's behavior.

The strongest architectural proposition of IEDCO Parallax is coordination. Scroll-driven depth should relate to the same motion decisions that govern entrances, exits, and feedback elsewhere in the interface. The opportunity is broader than moving a background image: a coordinated system can make scroll behavior easier to author, constrain the performance cost of that behavior, and preserve a consistent visual character.

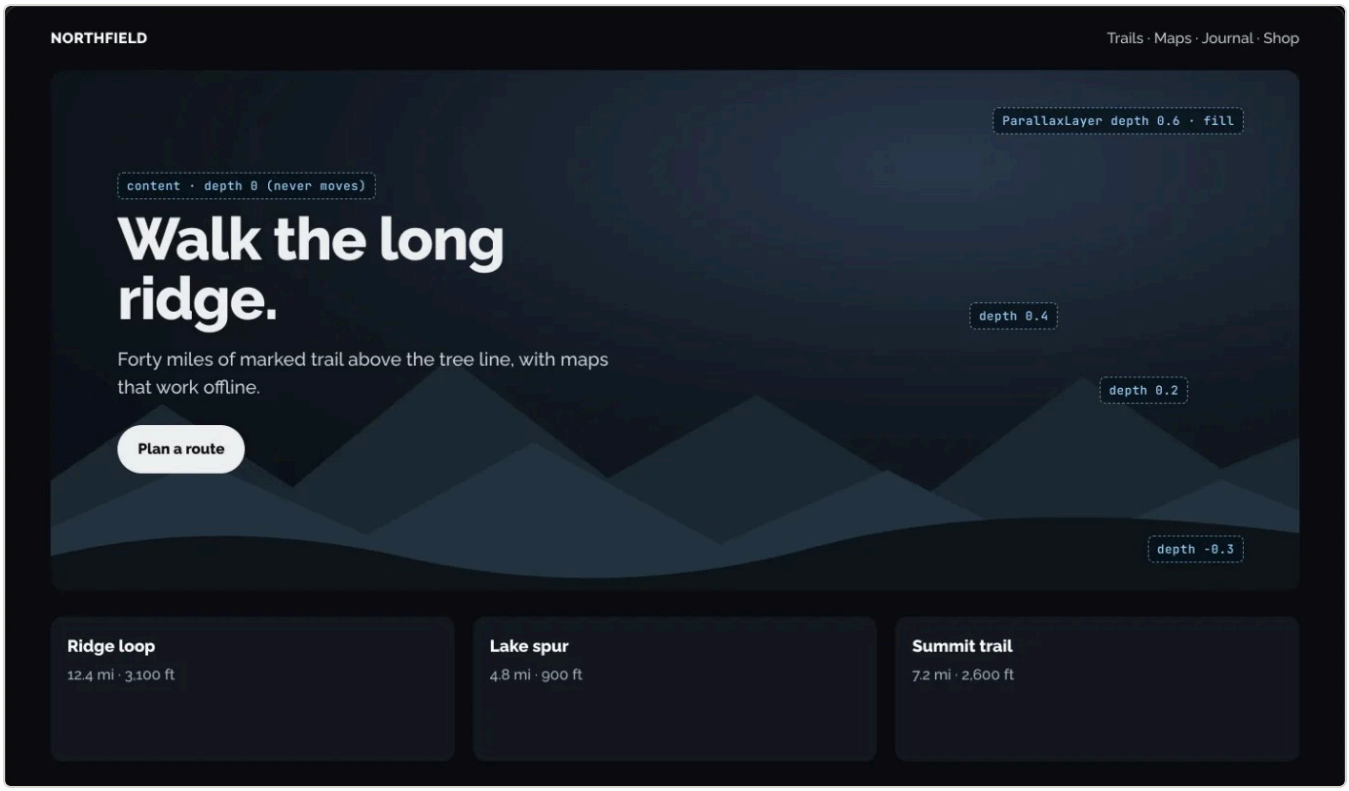


FIGURE 1 A demonstration scene from the product images. Labels mark each layer's depth: the content sits at depth 0 and never moves, background layers at 0.6, 0.4, and 0.2 sit farther away, and a near layer at −0.3 sits closer. Product image, IEDCO Parallax for React [18]. ADDED IN THIS EDITION

This paper is a design rationale combined with a research synthesis and is the companion to the IEDCO Labs paper on IEDCO Motion; it is not an independent evaluation of the product. The draft on which it is based described several scheduler, accessibility, and rendering features as recommendations that required verification before being represented as shipped capabilities. That distinction is kept: product facts are attributed to the product documentation, and recommendations are identified as recommendations. Section 2 defines terms and describes the product. Sections 3 through 5 explain why parallax works, how scroll progress differs from elapsed time, and what a lightweight implementation requires. Sections 6 and 7 describe the React and Angular integrations. Sections 8 through 10 cover coordination with IEDCO Motion, accessibility, and competitive position. Sections 11 through 13 present priorities for later versions, the remaining evidence gaps, and the boundaries of the evidence.

2. Conceptual foundations

2.1 Key terms

TERM	MEANING IN THIS PAPER
Layer	A visual plane that moves at its own rate as the page scrolls.
Scene	A section of a page together with its layers, assets, and content.
Progress	The position of the user within the scroll range of an effect, normalized from zero to one.
Range	How far a layer may move across the full progress of an effect.
Response	How directly a layer follows its target position; smoothing reduces directness.
Transition timing	How related, time-driven state changes begin and settle.
Scroll timeline	A timeline whose progress is set by scroll position rather than by elapsed time.
Reduced motion	The operating-system preference, exposed to CSS as `prefers-reduced-motion`, that asks for less movement.

2.2 Types of evidence

The sources cited in this paper differ in kind, and their weight differs accordingly. Product facts form a separate category, because the product's own documentation describes the product without independent verification.

CATEGORY	SOURCES	HOW THE EVIDENCE IS USED
Perception research	[1]	A psychophysical finding on motion parallax as a depth cue, assessed from the indexed abstract.

CATEGORY	SOURCES	HOW THE EVIDENCE IS USED
Platform specification	[2]	The definition of scroll-driven timelines. The document is a Working Draft, not a W3C Recommendation.
Engineering guidance	[3–7]	Browser-vendor guidance and demonstrations on rendering and scroll synchronization. One article dates from 2016.
Framework documentation	[8–12]	React and Angular guidance on references, effects, change detection, and lifecycle.
Accessibility guidance	[13, 14]	W3C explanations of a success criterion and a sufficient technique. These documents are informative, not normative.
Competitor documentation	[15–17]	Vendor descriptions of available capabilities, not independent comparative research.
Product documentation	[18]	The product's own description, published by IEDCO Labs. These facts have not been independently audited.

2.3

The product

According to its documentation, IEDCO Parallax is an add-on skill for IEDCO Motion. Claude adds layers that move at different speeds as the page scrolls, wires the layers into one section at a time, and measures each section frame by frame while scrolling the page. Backgrounds sit farther away and move more slowly than the page; small accents sit closer and move faster; text, buttons, and links never move. [18]

The documentation describes deliberately restrained defaults. On the Default preset, the farthest layer moves at most 32 px from its resting place, and half that distance on phones; a scene has at most three layers. Parallax reads Motion's configuration live, so the Motion Board's master switch, speed, and easing apply to both products. Version 1.1.0 adds a tuning panel with presets, sliders, and a scrub control, and the Angular edition, currently a beta, adds the same scroll depth to the Angular edition of Motion. The add-on uses no animation library, only two small files and some CSS on top of Motion. [18]

The documentation also lists what the current version does not include: pointer or hover tilt, device tilt, React Native, and parallax inside scrolling panels. [18]

3. Why parallax works

3.1 Relative movement as a depth cue

Parallax relies on relative movement. When layers respond differently to a change in viewpoint, a viewer can interpret the layers as occupying different depths. Perception research establishes the cue experimentally: Rogers and Graham found that relative image movement produced a reliable impression of depth in the absence of other depth cues, and in a stereoscopic matching task the perceived depth agreed closely with the degree of relative image displacement. [1]

A web page reproduces part of this relationship by translating layers at different rates as scrolling progresses. Controlled laboratory stimuli are not equivalent to a layered web page, however, and the experiments do not establish that layered scrolling improves usability or prescribe a sufficient travel distance.

3.2 A displacement model

A simple model expresses the displacement of a layer as scroll progress multiplied by the layer's movement range. For a section-based effect, scroll progress is normalized within a defined interval: progress is zero at the beginning of the interval and one at the end. A background layer might move through a smaller range than a foreground illustration, and the difference between their movements creates the depth cue.

The simple model assumes that each layer starts from zero offset and that progress maps linearly to movement. A general model adds a start offset and a mapping function, and an implementation must also define clamping, direction, and the treatment of nested scroll containers.

GENERAL FORM

```
displacement = start offset + mapping(progress) × range
```

3.3 Restraint and purpose

The effect depends on relationships rather than large distances. Subtle displacement can be sufficient when the composition already establishes foreground and background through scale, overlap, contrast, and cropping. The product's 32 px limit on the Default preset reflects that restraint. [18]

Parallax is most useful when it supports the communication of a page: separating visual planes, connecting sections, or guiding attention through a composed sequence. Parallax does not inherently improve comprehension, engagement, or conversion, and those outcomes require testing.

4. Scroll progress and elapsed time

4.1 Two kinds of timeline

Parallax introduces an important distinction between animation driven by time and animation driven by position. A conventional entrance animation progresses over a duration after an event. A scroll-linked effect progresses according to the user's location within a scroll range. The two mechanisms can share a visual language but should not be confused.

The web platform makes the same distinction. The W3C Scroll-driven Animations specification defines scroll progress timelines, whose progress is the scroll offset divided by the difference between the scrollable overflow size and the scroll container size, and view progress timelines, which track an element's progress through its scrollport. [2] The published document is a Working Draft dated June 6, 2023, so it does not by itself establish browser support or performance, and an animation attached to a scroll timeline may still apply easing.

4.2 Smoothing

If a user stops scrolling halfway through a section, a directly linked effect reflects that position, whereas a timed transition may continue until completion. Smoothing adds another layer of behavior: the displayed position approaches a target gradually and creates a softer response. Excessive smoothing can make the interface appear to lag behind the user.

4.3 Four quantities to keep separate

A coordinated system therefore distinguishes four quantities (Figure 2). Sharing motion settings should produce a family resemblance across these behaviors without forcing scroll position into a duration-based model.

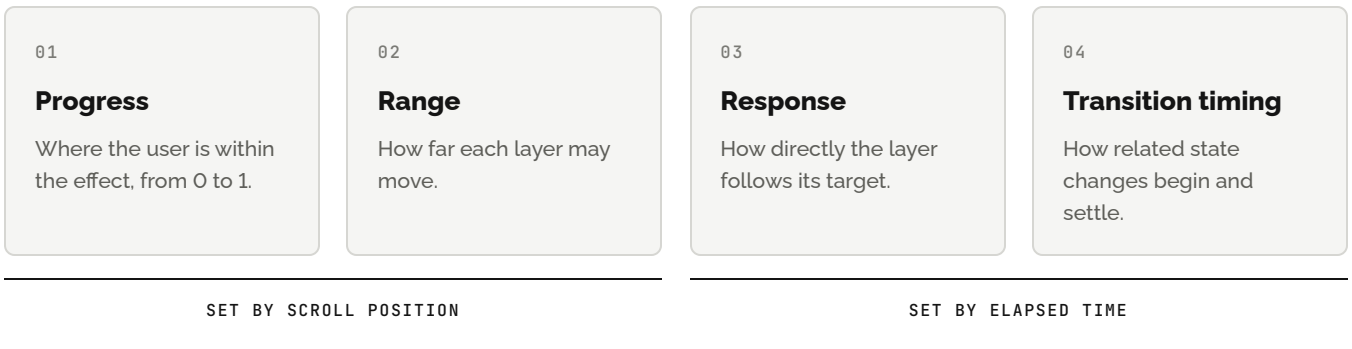


FIGURE 2 The four quantities a coordinated system keeps separate (Section 4.3), drawn from the text. Progress, range, and the target of the response follow scroll position; smoothing and transition timing follow elapsed time. ADDED IN THIS EDITION

5. A lightweight implementation strategy

5.1 Browser building blocks

A small parallax feature does not inherently require a broad animation engine. Browser APIs provide the building blocks for observing visibility, scheduling JavaScript work, measuring geometry, and applying visual changes, and CSS can handle the appropriate presentation rules. The efficiency of the result depends on the implementation.

5.2 Ordering reads and writes

For a JavaScript-driven version, a strong architecture collects input, schedules updates, reads the necessary measurements together, calculates positions, and then applies writes. Google's guidance on layout describes the cost of alternating geometry reads with style changes, which forces synchronous layout, and recommends reading style values before making style changes. [5] A `requestAnimationFrame` loop does not solve forced layout by itself: `requestAnimationFrame` aligns callbacks with rendering opportunities, but a read can still force layout if earlier work invalidated it.



SOURCE: WAGNER, LEWIS AND POLLARD, WEB.DEV [5], RETRIEVED OCT 22, 2026

FIGURE 3 The update order described in Section 5.2: geometry is read together before any transform is written. Alternating reads and writes, shown below the steps, is the pattern the layout guidance warns against. ADDED IN THIS EDITION

5.3 Rendering path

Transforms are generally a suitable starting point for layer displacement. Google's animation guidance recommends animating `transform` and `opacity` and checking any other property against the rendering pipeline. [7] An earlier Chrome article on parallax specifically warns against changing `background-position`, which causes the browser to repaint affected parts of the page on scroll, and against driving parallax from scroll events. [6] That article dates from 2016, and its browser-specific workaround is not a current universal recommendation. Large layers still carry memory and compositing costs, and excessive layer promotion can make performance worse.

5.4 Native scroll timelines

Browser-vendor material explains why native scroll timelines deserve evaluation. Scrolling is performed on a separate thread, so scroll events reach the main thread asynchronously, and animations that run on the main thread are subject to jank when other work competes for that thread. [3] CSS scroll-driven animations attach animations to scroll and view timelines declaratively. [4] These demonstrations are not IEDCO benchmarks, and native timelines do not make every animated property inexpensive or remove asset costs.

5.5 Idle work

A lightweight implementation should also avoid work when an effect is inactive. Offscreen sections, removed components, and settled smoothing calculations should not continue to consume processing time. These are engineering requirements for realizing the footprint advantage; avoiding an external package does not satisfy them automatically.

---

6. Integration with React

6.1 Rendering and continuous updates

React can define the page structure, the layer configuration, and the application state while a focused runtime handles continuous visual updates. React's documentation states that changing a ref does not trigger a re-render, which allows a runtime to keep element references and mutable animation data without scheduling a component render for each scroll increment. [8] Mutating a ref does not update a visible element by itself; a browser animation or a controlled DOM update must perform that work, and React and the runtime must not compete to own the same styles.

6.2 Geometry and cleanup

The integration must handle more than scrolling. Images loading, text wrapping, content expanding, and containers resizing can change the geometry on which an effect depends. Cleanup is equally important: route changes and component removal should release listeners, observers, and pending animation work. React's documentation describes Effects as the mechanism for synchronizing with external systems and makes cleanup part of that synchronization. [9]

A reusable React integration can make these responsibilities consistent across a page, which is a meaningful improvement over copying a custom scroll handler into every section.

7. Integration with Angular

7.1 Integration points

Angular requires an implementation that respects its templates, lifecycle, and change-detection model. Directives and services are possible integration points for registering layers and coordinating updates, and the appropriate choice depends on the released architecture and the supported Angular versions.

7.2 Change detection

The key performance principle is that continuous visual movement should not trigger unrelated application work. A browser-level position update should not require broad recalculation of the interface when the underlying application state has not changed. Angular's documentation identifies `requestAnimationFrame`, `setTimeout`, and `setInterval` as common causes of unnecessary change detection in zone-based applications and shows how to run such work outside the Angular zone. [10] Zoneless applications instead rely on explicit notifications to decide when to run change detection. [11] Running work outside the zone is therefore not a universal remedy; the correct boundary depends on the application's change-detection mode.

7.3 Lifecycle and rendering environments

Angular's cleanup mechanisms, `ngOnDestroy` and `DestroyRef`, should release resources when a view is destroyed. Rendering environments also need consideration: browser-dependent measurements should begin only when the required elements and APIs are available. Angular's render callbacks run after components have rendered to the DOM and do not run during server-side rendering or build-time pre-rendering. [12]

7.4 Shared meaning across editions

The React and Angular editions should share configuration meaning, motion character, and expected outcomes, while their internal integrations differ.

8. Coordination with IEDCO Motion

A page feels more intentional when its movement follows related rules. A restrained interface can become inconsistent if its parallax introduces large travel distances and prolonged settling. An energetic interface can feel disconnected if its scroll effects respond with a completely different character.

Coordination provides a way to manage these relationships. Motion preferences can influence the intensity and response of parallax while scroll-driven behavior retains its own controls. In the product, Parallax reads Motion's configuration live, so Motion's

master switch, speed, and easing apply to both. The documentation also states that layers hold still while a side panel pushes the page and then settle into place on Motion's curve, instead of jumping, when content loads above them. [18]

Whether a shared motion language improves the complete experience is a design proposition. Testing it would require comparing coordinated and independent scenes in task-based studies (Section 12).

9. Accessibility and comfort

9.1 Parallax as nonessential motion

A shared motion preference can guide both component animation and depth effects. Parallax often deserves stronger reduction than a small state transition, because parallax can introduce sustained, large-area movement. W3C's explanation of WCAG 2.2 Success Criterion 2.3.3, Animation from Interactions, names parallax scrolling as an animation that is often nonessential and notes that motion triggered by scrolling can cause vestibular reactions such as dizziness, nausea, and headaches. [13] The criterion is Level AAA and calls for interaction-triggered motion to be capable of being disabled unless the motion is essential; W3C lists the CSS `prefers-reduced-motion` media query as a sufficient technique for it. [14] Honoring the preference does not, by itself, establish conformance with WCAG.

Motion for React's documentation of its `useReducedMotion` hook gives turning off parallax motion as an example of adapting to the reduced-motion preference. [16]

9.2 The reduced experience

The reduced experience should preserve the composition, the content, and the interaction order, so that depth remains an enhancement to a usable page. The product documentation states that visitors who turn on the reduced-motion preference see every layer at rest with all content visible, and that version 1.1.0 keeps a layer from ever sitting over a button or link. [18] No source cited here establishes a comfortable travel distance or smoothing threshold.

10. Competitive position

The strongest advantage of IEDCO Parallax is a focused implementation connected to an existing motion language. For a page that needs a few coordinated depth effects, a separate animation engine may introduce unnecessary API surface and integration work, and a targeted system can reduce that burden while keeping the behavior understandable within the project.

ALTERNATIVE	STRENGTH OF THE ALTERNATIVE	CLAIMED ADVANTAGE OF IEDCO PARALLAX
Isolated parallax snippets	No added tooling.	Consistent configuration and a repeatable integration process.
Framework-specific packages	Native to one framework.	A shared way to discuss and configure behavior across React and Angular projects.
Broad animation libraries	Scroll-linked and scroll-triggered animation, parallax, and reduced-motion controls [15, 16]; tree shaking [17].	Scope: only the depth effects a page needs, on Motion's timings. Footprint advantage unmeasured.

These benefits do not establish universal performance superiority. An optimized library implementation may be efficient, and mature packages may offer substantial value for complex sequencing, pinning, gestures, and layout coordination. Motion for React, for example, states that it runs scroll-linked animations on the browser's native scroll timeline where possible. [15] Native execution and reduced-motion support are therefore not exclusive to IEDCO Parallax.

A fair comparison should measure equivalent scenes, including their images and layer costs. A small script that controls several oversized assets can still produce a heavy page.

11. Priorities for later versions

11.1 Establish scene-level performance budgets

Performance should be evaluated at the level users experience: the complete scene. Useful measurements include transferred assets, active layers, frame work, memory behavior, and responsiveness during scrolling. Tests should include mobile devices and pages that contain multiple effects.

11.2 Evaluate native scroll-driven animation

Later versions should evaluate CSS scroll-driven animation where browser support and effect requirements make it appropriate. A progressive approach could use native capabilities for suitable scenes and a controlled fallback elsewhere, with adoption following compatibility testing and behavioral comparison.

11.3 Improve geometry resilience

Effects should respond predictably to image loading, font changes, expanded content, responsive layouts, and nested scroll containers. A clear measurement and invalidation strategy would reduce jumps and stale positioning. Such a strategy needs more than one observer: `ResizeObserver` detects changes in an element's size but not every

change in its position, because an element can move when a preceding section expands, and `IntersectionObserver` can gate activity but is not a precise continuous scroll timeline.

11.4 Make transform ownership explicit

A layer may already use transforms for layout, hover behavior, or another animation, and independent systems that write the same transform can overwrite one another. Wrapper layers or composed transform values can give each behavior clear ownership and reduce integration conflicts.

11.5 Add composition safeguards

Authoring tools could warn when movement may expose empty edges, crop important content, or move interactive elements excessively. Responsive previews would help teams adjust travel ranges for narrower screens rather than simply shrinking a desktop composition. The 1.1.0 tuning panel, which previews layers live and can scrub to any point of the scroll, provides a starting point for such safeguards. [18]

11.6 Provide explicit comfort controls

Reduced-motion handling should be accompanied by clear control over intensity and smoothing. Teams should be able to preserve a static composition, reduce travel, or disable decorative depth without affecting access to content.

11.7 Publish an evidence-based comparison

A benchmark should compare equivalent IEDCO, native, and library-based implementations and report production footprint and rendering behavior separately from the authoring workflow. Such a benchmark would make the lightweight claim concrete and show where the focused approach offers the greatest benefit.

12. Remaining evidence gaps

External sources support the mechanics and requirements described here, but no cited source establishes that IEDCO Parallax is smaller, faster, or more comfortable than alternatives. The table lists each claim, the current evidence for it, and the step required to support it.

CLAIM OR FEATURE	CURRENT EVIDENCE	REQUIRED NEXT STEP
A consistent motion language improves the complete experience	Design proposition	Compare coordinated and independent scenes in task-based usability studies.

CLAIM OR FEATURE	CURRENT EVIDENCE	REQUIRED NEXT STEP
No general-purpose animation package is required	Product documentation [18]	Inspect distributed and generated code, imports, and the dependency graph.
Smaller production footprint	Unmeasured	Compare incremental compressed JavaScript and CSS, deferred chunks, and generated code.
Better scroll performance	Unmeasured	Compare identical assets, layer sizes, travel, devices, browsers, and main-thread load.
Smoothing improves comfort	No threshold established	Test direct and smoothed responses with users, keeping reduced-motion options.
Geometry stays correct after content changes	Partly documented [18]	Test image and font loading, layout shifts above the section, expansion, resizing, nested scrolling, and scroll restoration.
Inactive effects stop consuming work	Engineering requirement	Inspect callbacks, observers, scheduling, and settled or offscreen behavior.
Transform composition is reliable	Engineering requirement	Test hover, layout transforms, entrances, and parallax on the same scene.
React and Angular behave equivalently	Unverified	Run shared scenarios for progress, geometry, lifecycle, preferences, and navigation.
Improved conversion or engagement	Unsupported	A controlled, product-specific study; perceived depth is not evidence of either outcome.

13.

Methodological boundaries

The perception study [1] was assessed from its indexed abstract, because the publisher's full text could not be retrieved, and its laboratory stimuli differ from layered web pages. The scroll-driven animation specification [2] is a Working Draft. The browser-vendor material [3–7] consists of guidance and demonstrations rather than independent benchmarks, and the parallax article [6] is historical.

The framework and competitor documentation [8–12, 15–17] describes capabilities that may change after October 22, 2026, and should be checked again before any version-specific claim is published. The accessibility documents [13, 14] are informative explanations of a Level AAA criterion, and meeting that criterion is not the same as conforming to WCAG as a whole.

The product facts [18] come from IEDCO Labs, the publisher of both the product and this paper. They have not been independently reproduced, and no IEDCO Parallax code was audited for this paper.

14.

Conclusion

The strongest proposition of IEDCO Parallax is controlled depth that belongs to the rest of the interface. For React and Angular teams, a focused implementation can offer an attractive alternative to disconnected scroll effects or a broader animation dependency. The value of that implementation depends on coordination, predictable lifecycle behavior, responsive composition, and disciplined rendering.

Later versions should prioritize measurable performance, resilient geometry, accessibility, and stronger authoring safeguards. Those improvements would support the central promise of the product: depth that adds visual character while remaining manageable within an ordinary application.

Acknowledgments and contribution statement

Rob Vance is the author of the source manuscript. A research pass dated October 22, 2026 identified the supporting sources and suggested passages used in this edition. IEDCO Labs prepared this edition, which rewrites the manuscript as an impersonal research report; Appendix A records each substantive change. The editorial standards applied in the revision, namely clear pronoun referents, grammatical agreement, a formal introductory register, complete sentences, and precise terminology, follow Dr. Felicia Ledogar's annotations on the companion paper, Human writing voice. Dr. Ledogar did not review this manuscript, and this acknowledgment does not imply that she endorses the paper, the product, or the studies it cites.

References

Grouped by type of evidence. Every link was tested and every page read on October 22, 2026 (Appendix B).

PERCEPTION RESEARCH

[1] Rogers, B., & Graham, M. (1979). Motion Parallax as an Independent Cue for Depth Perception. Perception, 8(2), 125–134. Abstract record: PubMed 471676.
https://doi.org/10.1068/p080125
https://pubmed.ncbi.nlm.nih.gov/471676/

PLATFORM SPECIFICATION

[2] Birtles, B., Ballo, B., Quint, A., Gerchikov, O., Etemad, E. J., & Flack, R. (Eds.). (2023, June 6). Scroll-driven Animations. W3C Working Draft.
https://www.w3.org/TR/scroll-animations-1/

ENGINEERING GUIDANCE

[3] Hirota, Y. (2023). A case study on scroll-driven animations performance. Chrome for Developers. https://developer.chrome.com/blog/scroll-animation-performance-case-study

[4] Van Damme, B. (2023, May 5). Animate elements on scroll with Scroll-driven animations. Chrome for Developers. https://developer.chrome.com/docs/css-ui/scroll-driven-animations

[5] Wagner, J., Lewis, P., & Pollard, B. (2015; updated 2025). Avoid large, complex layouts and layout thrashing. web.dev, Google. https://web.dev/articles/avoid-large-complex-layouts-and-layout-thrashing

[6] Lewis, P., & Flack, R. (2016). Performant Parallaxing. Chrome for Developers. https://developer.chrome.com/blog/performant-parallaxing

[7] Basques, K., & Andrew, R. (n.d.). How to create high-performance CSS animations. web.dev, Google. Retrieved October 22, 2026. https://web.dev/articles/animations-guide

FRAMEWORK DOCUMENTATION

[8] React. (n.d.). useRef. React documentation. Retrieved October 22, 2026. https://react.dev/reference/react/useRef

[9] React. (n.d.). Synchronizing with Effects. React documentation. Retrieved October 22, 2026. https://react.dev/learn/synchronizing-with-effects

[10] Angular. (n.d.). Zone pollution. Angular documentation. Retrieved October 22, 2026. https://angular.dev/best-practices/zone-pollution

[11] Angular. (n.d.). Zoneless. Angular documentation. Retrieved October 22, 2026. https://angular.dev/guide/zoneless

[12] Angular. (n.d.). Component lifecycle. Angular documentation. Retrieved October 22, 2026. https://angular.dev/guide/components/lifecycle

ACCESSIBILITY GUIDANCE

[13] W3C Web Accessibility Initiative. (n.d.). Understanding Success Criterion 2.3.3: Animation from Interactions (WCAG 2.2). Retrieved October 22, 2026. https://www.w3.org/WAI/WCAG22/Understanding/animation-from-interactions.html

[14] W3C Web Accessibility Initiative. (n.d.). Technique C39: Using the CSS prefers-reduced-motion query to prevent motion (WCAG 2.2 Techniques). Retrieved October 22, 2026. https://www.w3.org/WAI/WCAG22/Techniques/css/C39

COMPETITOR DOCUMENTATION

[15] Motion. (n.d.). Scroll animation. Motion for React documentation. Retrieved October 22, 2026. https://motion.dev/docs/react-scroll-animations

[16] Motion. (n.d.). useReducedMotion. Motion for React documentation. Retrieved October 22, 2026. https://motion.dev/docs/react-use-reduced-motion

[17] Motion. (n.d.). Reduce bundle size. Motion for React documentation. Retrieved October 22, 2026. https://motion.dev/docs/react-reduce-bundle-size

PRODUCT DOCUMENTATION

[18] IEDCO Labs. (2026). IEDCO Parallax: React 1.1.0 and Angular 1.1.0 (beta). Labs, industrialethic.com. Retrieved October 22, 2026. The product image in Figure 1 is from the same entry.
https://industrialethic.com/#labs

Appendix A. Revision record

This edition rewrites the draft as an impersonal research report. Every claim, qualification, and caveat in the draft is retained, and the draft's product statements are now tied to the product documentation. The table lists each substantive change by draft section.

DRAFT SECTION	THIS EDITION	SUBSTANTIVE CHANGE
Abstract	Title page, 1	Rewritten as a formal abstract with keywords. The draft's note on unverified features kept in Section 1.
The problem	1	Restated as a problem definition. Figure 1 added from the product images.
None	2	Key terms, evidence types, and a product description from the product documentation [18] added.
Why parallax works	3	Perception research [1] added with its limits. The displacement model generalized with a start offset and mapping, as the research handoff recommended.
Scroll progress and elapsed time	4	The W3C specification [2] added, including its progress formula and Working Draft status. The four quantities drawn as Figure 2.
A lightweight implementation strategy	5	Layout and rendering guidance [5] to [7] and native-timeline material [3], [4] added. The limit of requestAnimationFrame stated. Figure 3 added.
How the approach fits React	6	Ref and Effect documentation [8], [9] added.
How the approach fits Angular	7	Zone, zoneless, and lifecycle documentation [10] to [12] added; running work outside the zone qualified by change-detection mode.
Why coordination with IEDCO Motion matters	8, 9	Split into coordination and accessibility. Product coordination facts [18] added. W3C guidance naming parallax [13], [14] added. Motion for React's parallax example now cited [16].
Where IEDCO Parallax offers an advantage	10	Comparison table added; competitor capabilities cited [15] to [17].
Priorities for later versions	11	Retained in full. Observer limits added to 11.3; the 1.1.0 tuning panel noted in 11.5.
None	12, 13	Evidence-gap table and methodological boundaries added from the research handoff.
Conclusion	14	Retained.
None	References	Eighteen references added; the draft named Motion's guidance without a link.

Editorial standards applied

The five annotations below were written by Dr. Felicia Ledogar on the companion paper. Each was applied to this manuscript as a general standard.

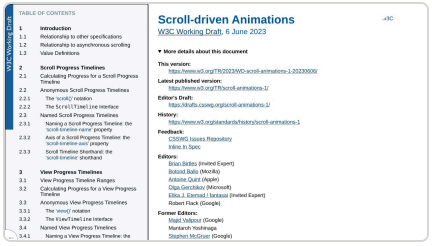
STANDARD	SOURCE ANNOTATION	APPLICATION IN THIS EDITION
Clear referents	“what does its refer to??”	“It,” “its,” and “this” replaced with the noun wherever more than one referent was possible.
Number agreement	“do not use they, it is plural”	Singular subjects such as “a layer” and “a user” are not followed by “they.”
Formal introduction	“intro??? too casual”	The draft’s short opening lines restated as a problem definition; headings formalized.
Complete sentences	“last sentence is not a sentence, there is no subject/noun”	Prose fragments expanded into complete sentences. Tables keep short entries.
Precise terminology	“contrastive?? or contrasting??”	Progress, range, response, and transition timing defined in Section 2.1 and used consistently.

Appendix B. Source verification

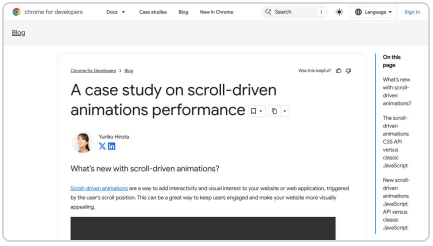
ADDED IN THIS EDITION This appendix is an IEDCO Labs production record. Every link in this edition is live and was tested on October 22, 2026, and each cited page was read to confirm that the page supports the sentence citing it. Reference [1] could not be opened by an automated browser: the publisher shows a bot check and PubMed rate-limits automated requests, so no capture is shown and the record was confirmed through Crossref and NCBI's E-utilities. The W3C pages answer automated requests but show a bot check to an automated browser; no bot check was attempted, and those captures were rendered from the fetched pages. Cookie banners were removed from the captures, not clicked.

REF.	HOST	RESULT
[1]	doi.org · pubmed	Publisher (SAGE) shows a bot check to automated browsers; PubMed rate-limits automated requests (HTTP 429). Record confirmed through Crossref (Perception 8(2), 125–134, Rogers and Graham) and through NCBI's E-utilities, which returned the abstract cited in Section 3.1.
[2]	w3.org	HTTP 200. Working Draft of June 6, 2023; the page defines scroll and view progress timelines and the progress formula. Captured from the fetched page, as for [13].
[3]	developer.chrome.com	HTTP 200. Title and author match; the page states scroll events reach the main thread asynchronously and main-thread animations are subject to jank.
[4]	developer.chrome.com	HTTP 200. Title, author and date (May 5, 2023) match.
[5]	web.dev	HTTP 200. Title, authors and dates match; the page advises reading style values before making style changes.
[6]	developer.chrome.com	HTTP 200. Title and authors match; last updated December 2, 2016. The page warns against scroll events and background-position for parallax.
[7]	web.dev	HTTP 200. Title and authors match; the page recommends transform and opacity.
[8]	react.dev	HTTP 200. The page states that changing a ref does not trigger a re-render.
[9]	react.dev	HTTP 200. The page covers Effects, external systems and cleanup.
[10]	angular.dev	HTTP 200. The page names requestAnimationFrame, setTimeout and setInterval and shows runOutsideAngular.
[11]	angular.dev	HTTP 200. The page describes notifications for zoneless change detection.
[12]	angular.dev	HTTP 200. The page lists ngOnDestroy, DestroyRef and render callbacks, which do not run during server rendering.

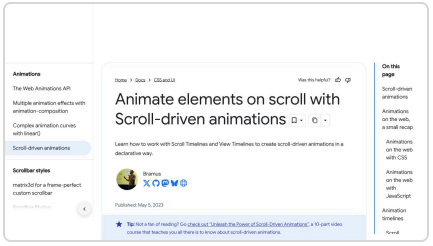
REF.	HOST	RESULT
[13]	w3.org	HTTP 200 to curl. The page names parallax scrolling as often nonessential and identifies 2.3.3 as Level AAA. A real browser receives a bot check, so the capture is rendered from the fetched page.
[14]	w3.org	HTTP 200 to curl. The technique is listed as sufficient for 2.3.3. Captured from the fetched page, as for [13].
[15]	motion.dev	HTTP 200. The page covers scroll-linked animation and parallax and states use of the native scroll timeline where possible.
[16]	motion.dev	HTTP 200. The page gives turning off parallax motion as a reduced-motion example.
[17]	motion.dev	HTTP 200. The page explains tree shaking of Motion for React.
[18]	industrialethic.com	HTTP 200. The live Labs page carries the product facts cited (32 px limit, three layers, reduced-motion behavior, 1.1.0 tuning panel).



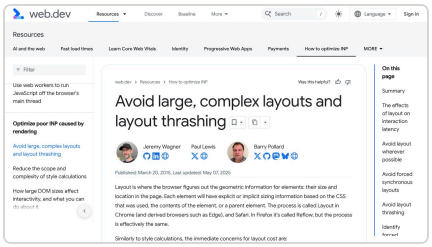
[2] W3.ORG · OCT 22, 2026



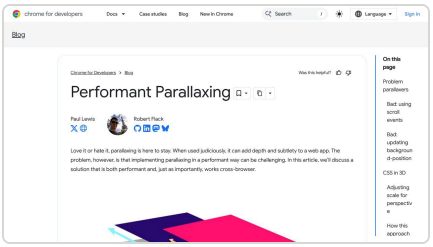
[3] DEVELOPER.CHROME.COM · OCT 22, 2026



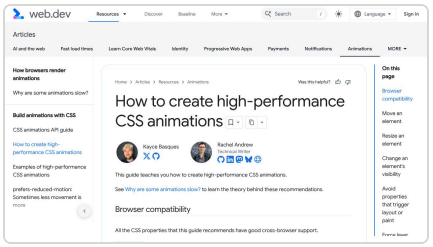
[4] DEVELOPER.CHROME.COM · OCT 22, 2026



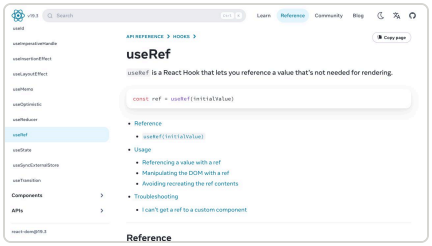
[5] WEB.DEV · OCT 22, 2026



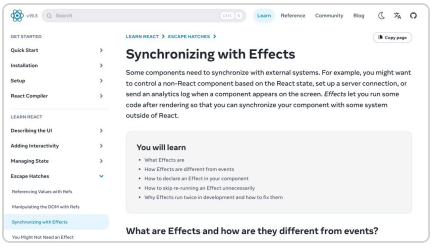
[6] DEVELOPER.CHROME.COM · OCT 22, 2026



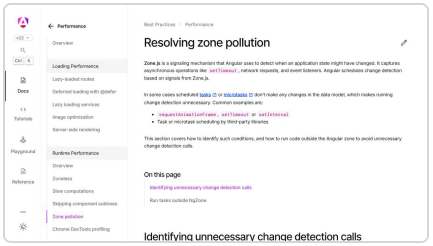
[7] WEB.DEV · OCT 22, 2026



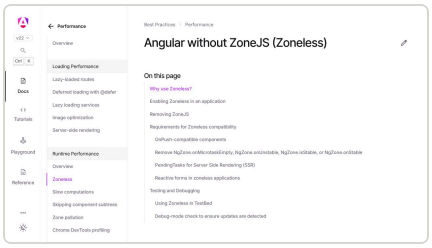
[8] REACT.DEV · OCT 22, 2026



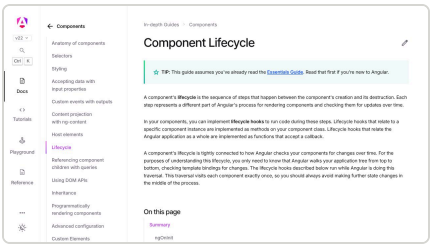
[9] REACT.DEV · OCT 22, 2026



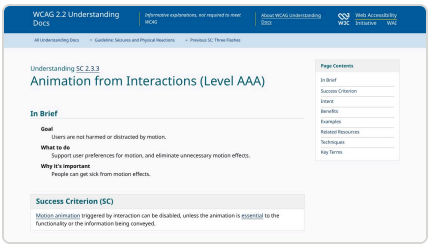
[10] ANGULAR.DEV · OCT 22, 2026



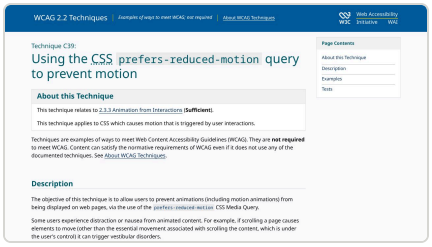
[11] ANGULAR.DEV · OCT 22, 2026



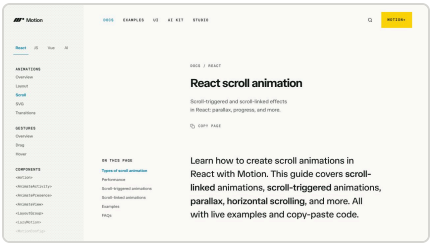
[12] ANGULAR.DEV · OCT 22, 2026



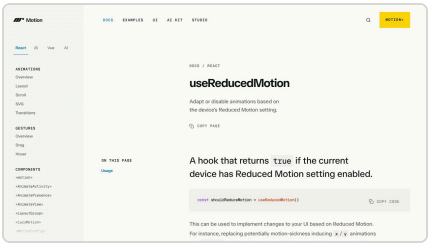
[13] W3.ORG · OCT 22, 2026



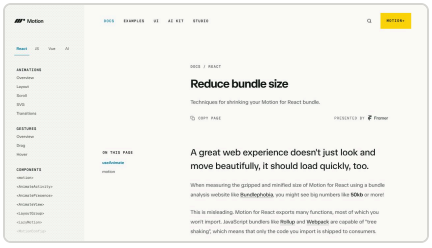
[14] W3.ORG · OCT 22, 2026



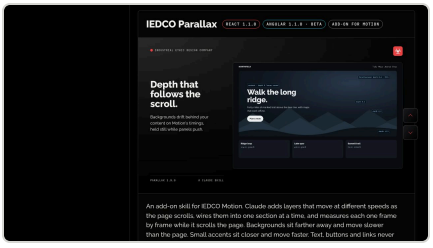
[15] MOTION.DEV · OCT 22, 2026



[16] MOTION.DEV · OCT 22, 2026



[17] MOTION.DEV · OCT 22, 2026



[18] INDUSTRIALETHIC.COM · OCT 22, 2026