

Exhibit G

Valve's Accused VR Instrumentalities infringe the following asserted claims of U.S. Patent No. 11,175,738 to Khoshkava, et al. (the "'738 patent") in at least the following exemplary ways. To the extent that the cited documentation relating to Valve's Accused VR Instrumentalities does not disclose all features, methods or operations required by the asserted claims, such non-disclosed features, methods or operations are inherently or equivalently practiced. Immersion's preliminary infringement contentions are based on information presently available to Immersion. Based upon that information, Valve's deployment, operation, maintenance, testing, use, sale, and importation of the Accused VR Instrumentalities as described below infringes one or more claims of the '738 Patent either literally or under the doctrine of equivalents. Immersion reserves the right to provide further details regarding Valve's infringement, whether that infringement is literal or under the doctrine of equivalents, as discovery is provided by Valve, and to the fullest extent permissible under the Federal and Local Rules, and any Order of this Court. The specific components, systems, and constructs identified in this chart are for exemplary purposes only and Immersion reserves all rights to supplement as additional components, systems, and constructs become known through discovery, as well as after Valve produces documents and source code and/or the Court construes any terms from the claims of the '738 Patent.

For example, Valve provides guides for developing games and applications with the Unreal, Unity, and Source engines for the Accused VR Instrumentalities, which are subject to Valve's control and used for Valve's benefit. In addition, certain developers may develop games and applications with different engines that nonetheless must interoperate with the Accused VR Instrumentalities and are subject to Valve's control and used for Valve's benefit. Regardless of which engine is deployed, the Accused VR Instrumentalities infringe the asserted claims, as identified in the examples below.

Valve directly infringes the claims of the '738 Patent in the exemplary ways detailed below. Valve indirectly infringes the claims of the '738 Patent to the extent Valve's users are direct infringers of the claims of the '738 Patent as referenced below, and their acts of direct infringement are use of the Accused VR Instrumentalities in an infringing manner as described below. With knowledge of infringement of the claims of the '738 Patent going back to at least the filing of the Complaint in this matter, Valve has taken and continues to take active steps to induce infringement of the claims of the '738 Patent by its users. Such active steps include, but are not limited to, making and selling the Accused VR Instrumentalities, encouraging third parties to develop infringing features for the Accused VR Instrumentalities, providing its users with instructions on use of the accused features of Accused VR Instrumentalities.

In addition, Valve infringes by supplying or causing to be supplied in or from the United States all or a substantial portion of the components of the Accused VR Instrumentalities, including software, in such manner as to actively induce the combination of such components outside of the United States in a manner that would infringe the patent if such combination occurred within the United

States. Valve further also causes to be supplied in or from the United States components of the Accused VR Instrumentalities especially made for use in the Accused VR Instrumentalities, including without limitation software, that are not staple articles of commerce suitable for substantial noninfringing use, knowing that such component is so made or adapted and intending that such component will be combined outside of the United States in a manner that would infringe the patent if such combination occurred within the United States.

Claim 1	Valve's Accused Instrumentalities
1[P] A device comprising:	To the extent the preamble is determined to be limiting, Valve's Accused VR Instrumentalities include the claimed device that meets the limitations below.
1[A] a proximity sensor capable of detecting a non-contact interaction with a touch-sensitive device and outputting a first sensor signal;	Valve's Accused VR Instrumentalities include a proximity sensor capable of detecting a non-contact interaction with a touch-sensitive device and outputting a first sensor signal. For example Valve Index Controllers, also known as Knuckles,¹ use sensors to track finger position:² Sensor fusion Each controller uses 87 sensors to track hand position, finger position, motion, and pressure to determine user intent. All of these signals, combined with fine-tuned software and algorithms, give us a better understanding of how a player is holding and using the controllers.

¹ See https://www.ifixit.com/Device/Valve_Index_Controller (last accessed May 10, 2023).

² See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 1	Valve's Accused Instrumentalities
	Further, Valve documentation notes that, “[t]o read controller input state, applications provide a list of the ‘actions’ that represent the operations a user can perform. SteamVR then binds those actions to actual inputs on a game/vr controller,” or the specific input for the specific device the user is using:³ Input Overview To read controller input state in OpenVR, an application provides a list of the "actions" that represent the operations a user can perform in the application. SteamVR then binds those actions to actual inputs on a game/vr controller, or more specifically input components provided for the input devices that the user is using. Application developers can provide default bindings for any controller types they like. Users are able to modify those bindings or create new bindings for new devices. Users can also select bindings that were shared by other users. Additionally, the Skeletal Input system allows drivers for each type of controller to provide applications with an animated skeleton of the user's hand to the best level of fidelity that its sensors are able to detect. Skeletal data can be bound to input actions, just like other input:⁴ Overview The Skeletal Input system allows drivers for each type of controller to provide applications with an animated skeleton of the user's hand to the best level of fidelity that its sensors are able to detect. Developers can then reduce their need to create custom hand animations for each controller they plan to support, and can instead use the animation provided by the current driver in conjunction with their own game-specific animations. Skeletal data can be bound to input actions, just like other input. The animation can be requested to be as accurate as possible (which will often mean that the range of motion of the For example, the Valve Index controllers provide the “Partial” level of fidelity that measures finger curl:⁵

³ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Input#input-overview> (last accessed May 10, 2023).

⁴ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

⁵ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

Claim 1	Valve's Accused Instrumentalities
	Skeletal Tracking Level Different controllers have different abilities to track the individual joints of your fingers. Here we provide an overview value that should give you a general idea what level of fidelity your current controller has. • Estimated: Body part location can't be directly determined by the device. Any skeletal pose provided by the device is estimated based on the active buttons, triggers, joysticks, or other input sensors. Examples include the Vive Controller and gamepads. • Partial: Body part location can be measured directly but with fewer degrees of freedom than the actual body part. Certain body part positions may be unmeasured by the device and estimated from other input data. Examples include Knuckles or gloves that only measure finger curl • Full: Body part location can be measured directly throughout the entire range of motion of the body part. Examples include hi-end mocap systems, or gloves that measure the rotation of each finger segment. Further, SteamVR Skeleton Input provides an API to get per joint data for all of these different types of devices.⁶ For example, this API can provide curl information for the index finger:⁷ Finger Curls For some things it may be more useful to access a summary of how curled the finger is as opposed to the position and rotation for 4 joints on each finger. These values range from 0 to 1, with 1 being fully curled. You can access curl information at <code>skeletonAction.fingerCurls[]</code> as an array, or individually named at <code>skeletonAction.indexCurl</code>, <code>skeletonAction.middleCurl</code>, <code>skeletonAction.RingCurl</code>, <code>skeletonAction.pinkyCurl</code>, and <code>skeletonAction.thumbCurl</code>. Using the Skeletal Input API requires adding a “skeletal action” entry for each hand to the actions section of an application's action manifest file. Bindings then need to be added for these actions to each controller:⁸

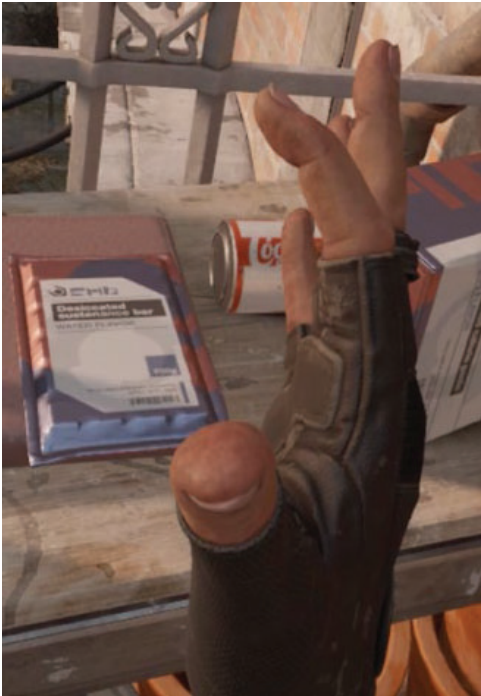
⁶ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

⁷ See https://valvesoftware.github.io/steamvr_unity_plugin/articles/Skeleton-Input.html#finger-curls (last accessed May 10, 2023).

⁸ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

Claim 1	Valve's Accused Instrumentalities
	Add an Action and Controller Binding To use the Skeletal Input API, first add a "skeletal action" entry for each hand to the actions section of your application's action manifest file. Make sure that the 'type' is skeleton, and that it has an additional 'skeleton' field to define which skeleton the action should use. The choices for skeleton are either "/skeleton/hand/left" or "/skeleton/hand/right". For example: <pre data-bbox="516 456 1761 641"> { "name": "/actions/demo/in/lefthand_anim", "type": "skeleton", "skeleton": "/skeleton/hand/left" }</pre> Next, you need to add a binding for this action to each controller. You can do this through the SteamVR binding UI (preferred). The <i>Half-Life: Alyx</i> installation includes json files under “Half-Life Alyx/games/hlvr/cfg” that confirm this configuration. For example, the action manifest file, “actions.json,” includes the following skeletal action entries: <pre data-bbox="516 898 1453 1185"> 215 { 216 "name" : "/actions/default/in/HandSkeletonLeft", 217 "skeleton" : "/skeleton/hand/left", 218 "type" : "skeleton" 219 }, 220 { 221 "name" : "/actions/default/in/HandSkeletonRight", 222 "skeleton" : "/skeleton/hand/right", 223 "type" : "skeleton" 224 },</pre> The Knuckles bindings file, “bindings_knuckles.json,” includes the bindings of these actions:

Claim 1	Valve's Accused Instrumentalities
	<pre> 26 "skeleton" : [27 { 28 "output" : "/actions/default/in/handskeletonleft", 29 "path" : "/user/hand/left/input/skeleton/left" 30 }, 31 { 32 "output" : "/actions/default/in/handskeletonright", 33 "path" : "/user/hand/right/input/skeleton/right" 34 } 35], </pre> The bindings file includes additional bindings for the actions for each individual finger curl: <pre> 100 { 101 "inputs" : { 102 "pull" : { 103 "output" : "/actions/default/in/fingercurlindex" 104 } 105 }, 106 "mode" : "trigger", 107 "path" : "/user/hand/right/input/finger/index" 108 }, 109 { 110 "inputs" : { 111 "pull" : { 112 "output" : "/actions/default/in/fingercurlmiddle" 113 } 114 }, 115 "mode" : "trigger", 116 "path" : "/user/hand/right/input/finger/middle" 117 }, 118 { 119 "inputs" : { 120 "pull" : { 121 "output" : "/actions/default/in/fingercurlring" 122 } 123 }, 124 "mode" : "trigger", 125 "path" : "/user/hand/right/input/finger/ring" 126 }, </pre>

Claim 1	Valve's Accused Instrumentalities
	Further, the figure below is a screenshot taken from a Valve Index while testing this functionality. As just one example, curling an index finger towards the Index Controller's trigger causes the virtual finger to also curl, indicating that Valve's Accused VR Instrumentalities detect, by a proximity sensor, a non-contact interaction with a touch-sensitive device:  As another example in <i>Vertigo 2</i>⁹, curling an index finger towards the Index Controller's trigger causes the virtual finger to also curl, indicating that Valve's Accused VR Instrumentalities detect, by a proximity sensor, a non-contact interaction with a touch-sensitive device:

⁹ See https://store.steampowered.com/app/843390/Vertigo_2/.

Claim 1	Valve's Accused Instrumentalities
	
1[B] a touch sensor capable of detecting a touch with the touch-sensitive input device and outputting a second sensor signal;	Valve's Accused VR Instrumentalities include a touch sensor capable of detecting a touch with the touch-sensitive input device and outputting a second sensor signal. For example, Valve Index Controllers, also known as Knuckles,¹⁰ use sensors to track pressure:¹¹

¹⁰ See https://www.ifixit.com/Device/Valve_Index_Controller (last accessed May 10, 2023).

¹¹ See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 1	Valve's Accused Instrumentalities
	Sensor fusion Each controller uses 87 sensors to track hand position, finger position, motion, and pressure to determine user intent. All of these signals, combined with fine-tuned software and algorithms, give us a better understanding of how a player is holding and using the controllers. Further, the figure below is a screenshot taken from a Valve Index while testing this functionality. As just one example, touching the Index Controller's trigger causes the virtual finger to partially ball into the virtual hand, indicating that Valve's Accused VR Instrumentalities detect a touch on the touch-sensitive device:

Claim 1	Valve's Accused Instrumentalities
	 Note that the above hand pose differs from when a user pulls the trigger, which causes the finger to fully ball into the virtual hand, further indicating that Valve's Accused VR Instrumentalities detect a touch on the touch-sensitive device, as distinct from a trigger pull:

Claim 1	Valve's Accused Instrumentalities
	 As just another example in <i>Vertigo 2</i>, touching the Index Controller's trigger causes the virtual finger to partially ball into the virtual hand, indicating that Valve's Accused VR Instrumentalities detect a touch on the touch-sensitive device:

Claim 1	Valve's Accused Instrumentalities
	
1[C] a haptic output device configured to receive a haptic output signal and output a haptic effect in response to the haptic output signal; and	Valve's Accused VR Instrumentalities include a haptic output device configured to receive a haptic output signal and output a haptic effect in response to the haptic output signal. For example, the Valve Index Controllers include an HD LRA for Haptics:¹²

¹² See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 1	Valve's Accused Instrumentalities								
	Tech specs <table><tr><th>Inputs</th><th>Ergonomics</th><th>Connections</th><th>Haptics</th></tr><tr><td>A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU</td><td>Adjustable strap with pivot, antimicrobial tech fabric</td><td>USB-C, 2.4GHz Wireless</td><td>HD LRA</td></tr></table>	Inputs	Ergonomics	Connections	Haptics	A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless	HD LRA
Inputs	Ergonomics	Connections	Haptics						
A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless	HD LRA						
1[D] a processor configured to:	Valve's Accused VR Instrumentalities include a processor configured to meet the limitations below. See below. Immersion believes this limitation is literally met. Discovery has not opened however, to the extent that Valve identifies the circuitry within and argues it is not a processor and is instead, for example, a SOC or ASIC, they are both meet the claim limitation of processor as they are configured to do 1[E], 1[F], and 1[G]. This limitation is also met under the doctrine of equivalents. Under the function, way, result analysis, the function here is to process signals, such as those identified in 1[E], 1[F], 1[G]. Here, a SoC and an ASIC also process signals. A SoC processes signals, typically by specific integrated components. Likewise, an ASIC is hardwired to perform a specific purpose of processing signals. A person of ordinary skill in the art would recognize a SoC or ASIC as performing substantially the same function. Under the way, the way is by being configured (for example by having instructions) such that it can perform the limitations in 1[E], 1[F], and 1[G]. Here, a SoC also is configured (by having instructions) such that it would perform the same limitations. A ASIC is also configured (typically hardwired logic) such that it would perform the same limitations. A person of ordinary skill in the art would recognize a SoC or ASIC as performing substantially the same way. Under the result, the result is a processor configured in a manner such that it performs as described in 1[E], 1[F], and 1[G]. As explained below for those limitations, the same result occurs by the circuitry, whether it is								

Claim 1	Valve’s Accused Instrumentalities								
	a processor, SoC, or ASIC. A person of ordinary skill in the art would recognize a SoC or ASIC as performing in such a manner as substantially the same result. Under the insubstantial differences test, a person of ordinary skill in the art would view the differences between a processor and a SoC or ASIC as insubstantial. A SoC or ASIC do not affect the overall function or result.								
1[E] receive the first sensor signal and the second sensor signal;	Valve’s Accused VR Instrumentalities include a processor configured to receive the first sensor signal and the second sensor signal. The output to the display for the uses cases in Limitations 1[A] and 1[B] indicate that the processor is configured to receive the first sensor signal and the second sensor signal.								
1[F] generate a haptic output signal based at least in part on the first and second sensor signals; and	Valve’s Accused VR Instrumentalities include a processor configured to generate a haptic output signal based at least in part on the first and second sensor signals. For example, the Valve Index Controllers include an HD LRA for Haptics:¹³ Tech specs <table><thead><tr><th>Inputs</th><th>Ergonomics</th><th>Connections</th><th>Haptics</th></tr></thead><tbody><tr><td>A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU</td><td>Adjustable strap with pivot, antimicrobial tech fabric</td><td>USB-C, 2.4GHz Wireless</td><td>HD LRA</td></tr></tbody></table> Further, the “bindings_knuckles.json” configuration file includes bindings for haptics actions:	Inputs	Ergonomics	Connections	Haptics	A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless	HD LRA
Inputs	Ergonomics	Connections	Haptics						
A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless	HD LRA						

¹³ See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 1	Valve's Accused Instrumentalities
	<pre> 4 "bindings" : { 5 "/actions/default" : { 6 "haptics" : [7 { 8 "output" : "/actions/default/in/hapticvibration", 9 "path" : "/user/hand/left/output/haptic" 10 }, 11 { 12 "output" : "/actions/default/in/hapticvibration", 13 "path" : "/user/hand/right/output/haptic" 14 } 15], </pre> Further, the figures below are screenshots taken from a Valve Index while testing this functionality. As just one example, as mentioned in Limitation 1A, curling an index finger towards the trigger causes a virtual finger to curl, as shown in the left figure below, which are screenshots taken from a Valve Index while testing this functionality. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects occur. Additionally, as mentioned in Limitation 1B, touching the trigger also causes the virtual finger to curl, as shown in the right figure below. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects also occur: 

Claim 1	Valve's Accused Instrumentalities
	Note that pulling the trigger results in the virtual hand grabbing the object, further indicating that Valve's Accused VR Instrumentalities independently track proximity to, touching, and pulling the trigger. Based on this behavior and on information and belief, Valve's Accused VR Instrumentalities generate a haptic output signal based at least in part on the first and second sensor signals.  As another example in <i>Vertigo 2</i>, as mentioned in Limitation 1A, curling an index finger towards the trigger causes a virtual finger to curl, as shown in the left figure below, which are screenshots taken from a Valve Index while testing this functionality. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects occur. Additionally, as mentioned in Limitation 1B, touching the trigger also causes the virtual finger to curl, as shown in the right figure below. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects also occur:

Claim 1	Valve's Accused Instrumentalities
	Based on this behavior and on information and belief, Valve's Accused VR Instrumentalities generate a haptic output signal based at least in part on the first and second sensor signals. The Accused Instrumentalities literally infringe claim 1. If Valve asserts that this limitation is not literally present in Valve's Accused VR Instrumentalities as set forth above, it is found in Valve's Accused VR Instrumentalities under the doctrine of equivalents because any differences between such claim elements and Valve's Accused VR Instrumentalities are insubstantial and/or the accused products perform substantially the same function, in substantially the same way to achieve substantially the same result as the corresponding claim elements. The claimed function is to generate a haptic output signal based at least in part on the first and second sensor signals. The claimed way is to base the haptic output signal at least in part on the first and second sensor signals. The claimed result is an outputted haptic effect, based at least in part on the first and second sensor signals. The function in Valve's Accused VR Instrumentalities is to process both a signal from the proximity sensor and a signal from the touch sensor, and generate a haptic effect in response to the described use case. On information and belief, the way the Valve's Accused VR Instrumentalities achieve this function is by receiving both the proximity signal and the touch signal. The observed result is the same as the claimed result.

Immersion's Preliminary Infringement Contentions (10/20/2023)

Claim 1	Valve's Accused Instrumentalities
	Under the insubstantial differences test, a person of ordinary skill in the art would view the differences as insubstantial and not affecting the overall function or result.
1[G] transmit the haptic output signal to the haptic output device.	Valve's Accused VR Instrumentalities include a processor configured to transmit the haptic output signal to the haptic output device. See limitation 1[C]. Additionally, testing confirms the haptic effect for the use cases in Limitation 1[F], which demonstrates that Valve's Accused VR Instrumentalities transmit the haptic output signal to the haptic output device.

Claim 2	Valve's Accused Instrumentalities
2. The device of claim 1, wherein the proximity sensor is integrated into the touch-sensitive device.	Valve's Accused VR Instrumentalities include the device of claim 1, wherein the proximity sensor is integrated into the touch-sensitive device. Both of the uses cases in Limitations 1[A] and 1[B] involve the trigger input for the Valve Index Controller, which indicates that the proximity sensor is integrated into the touch-sensitive device. Further, each controller uses 87 sensors to track hand position, finger position, motion, and pressure to determine user intent.¹⁴ Sensor fusion Each controller uses 87 sensors to track hand position, finger position, motion, and pressure to determine user intent. All of these signals, combined with fine-tuned software and algorithms, give us a better understanding of how a player is holding and using the controllers.

¹⁴ See <https://www.valvesoftware.com/en/index/controllers>.

Claim 3	Valve's Accused Instrumentalities
3. The device of claim 1, wherein the proximity sensor comprises one of a capacitive sensor or an optical sensor.	Valve's Accused VR Instrumentalities include the device of claim 1, wherein the proximity sensor comprises one of a capacitive sensor or an optical sensor. For example, controllers in Valve's Accused VR Instrumentalities include a capacitive sensor: Throwing Enabling natural and accurate open-handed throwing was a high priority for the team, and we used every sensor at our disposal. Optical and motion sensors provide velocity and trajectory, capacitive and force sensors help determine the intended release point, and our fine-tuned algorithms and software bring it all together into a usable input / output. The end result is an intuitive and natural-feeling throw.

Claim 6	Valve's Accused Instrumentalities							
6. The device of claim 5, wherein the integrated sensor further comprises the haptic output device.	Valve's Accused VR Instrumentalities include the device of claim 5, wherein the integrated sensor further comprises the haptic output device.							
	For example, the Valve Index Controllers include an HD LRA for Haptics: ¹⁵							
	Tech specs <table><tr><th>Inputs</th><th>Ergonomics</th><th>Connections</th><th>Haptics</th></tr><tr><td>A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU</td><td>Adjustable strap with pivot, antimicrobial tech fabric</td><td>USB-C, 2.4GHz Wireless</td><td>HD LRA</td></tr></table>	Inputs	Ergonomics	Connections	Haptics	A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless
Inputs	Ergonomics	Connections	Haptics					
A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless	HD LRA					

¹⁵ See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 14	Valve's Accused Instrumentalities
14. The device of claim 1, wherein the touch sensor is capable of detecting a pressure and wherein the haptic output signal is based at least in part on the detected pressure.	Valve's Accused VR Instrumentalities include the device of claim 1, wherein the touch sensor is capable of detecting a pressure and wherein the haptic output signal is based at least in part on the detected pressure. For example, Valve Index Controllers, also known as Knuckles,¹⁶ use sensors to track pressure:¹⁷ Sensor fusion Each controller uses 87 sensors to track hand position, finger position, motion, and pressure to determine user intent. All of these signals, combined with fine-tuned software and algorithms, give us a better understanding of how a player is holding and using the controllers.
Claim 15	Valve's Accused Instrumentalities
15[P]. A method comprising:	To the extent the preamble is determined to be limiting, Valve's Accused VR Instrumentalities perform the claimed method as explained below.

¹⁶ See https://www.ifixit.com/Device/Valve_Index_Controller (last accessed May 10, 2023).

¹⁷ See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities
15[A] detecting, by a proximity sensor, a non-contact interaction with a touch-sensitive device;	Valve's Accused VR Instrumentalities detect, by a proximity sensor, a non-contact interaction with a touch-sensitive device. For example Valve Index Controllers, also known as Knuckles,¹⁸ use sensors to track finger position:¹⁹ Further, Valve documentation notes that, “[t]o read controller input state, applications provide a list of the ‘actions’ that represent the operations a user can perform. SteamVR then binds those actions to actual inputs on a game/vr controller,” or the specific input for the specific device the user is using:²⁰

¹⁸ See https://www.ifixit.com/Device/Valve_Index_Controller (last accessed May 10, 2023).

¹⁹ See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

²⁰ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Input#input-overview> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities
	Input Overview To read controller input state in OpenVR, an application provides a list of the "actions" that represent the operations a user can perform in the application. SteamVR then binds those actions to actual inputs on a game/vr controller, or more specifically input components provided for the input devices that the user is using. Application developers can provide default bindings for any controller types they like. Users are able to modify those bindings or create new bindings for new devices. Users can also select bindings that were shared by other users. Additionally, the Skeletal Input system allows drivers for each type of controller to provide applications with an animated skeleton of the user's hand to the best level of fidelity that its sensors are able to detect. Skeletal data can be bound to input actions, just like other input:²¹ Overview The Skeletal Input system allows drivers for each type of controller to provide applications with an animated skeleton of the user's hand to the best level of fidelity that its sensors are able to detect. Developers can then reduce their need to create custom hand animations for each controller they plan to support, and can instead use the animation provided by the current driver in conjunction with their own game-specific animations. Skeletal data can be bound to input actions, just like other input. The animation can be requested to be as accurate as possible (which will often mean that the range of motion of the For example, the Valve Index controllers provide the "Partial" level of fidelity that measures finger curl:²²

²¹ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

²² See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities
	Skeletal Tracking Level Different controllers have different abilities to track the individual joints of your fingers. Here we provide an overview value that should give you a general idea what level of fidelity your current controller has. • Estimated: Body part location can't be directly determined by the device. Any skeletal pose provided by the device is estimated based on the active buttons, triggers, joysticks, or other input sensors. Examples include the Vive Controller and gamepads. • Partial: Body part location can be measured directly but with fewer degrees of freedom than the actual body part. Certain body part positions may be unmeasured by the device and estimated from other input data. Examples include Knuckles or gloves that only measure finger curl • Full: Body part location can be measured directly throughout the entire range of motion of the body part. Examples include hi-end mocap systems, or gloves that measure the rotation of each finger segment. Further, SteamVR Skeleton Input provides an API to get per joint data for all of these different types of devices.²³ For example, this API can provide curl information for the index finger:²⁴ Finger Curls For some things it may be more useful to access a summary of how curled the finger is as opposed to the position and rotation for 4 joints on each finger. These values range from 0 to 1, with 1 being fully curled. You can access curl information at <code>skeletonAction.fingerCurls[]</code> as an array, or individually named at <code>skeletonAction.indexCurl</code>, <code>skeletonAction.middleCurl</code>, <code>skeletonAction.RingCurl</code>, <code>skeletonAction.pinkyCurl</code>, and <code>skeletonAction.thumbCurl</code>. Using the Skeletal Input API requires adding a “skeletal action” entry for each hand to the actions section of an application’s action manifest file. Bindings then need to be added for these actions to each controller:²⁵

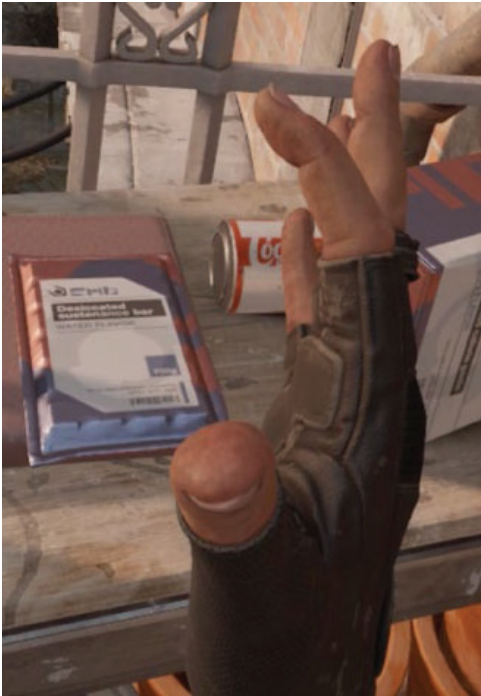
²³ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

²⁴ See https://valvesoftware.github.io/steamvr_unity_plugin/articles/Skeleton-Input.html#finger-curls (last accessed May 10, 2023).

²⁵ See <https://github.com/ValveSoftware/openvr/wiki/SteamVR-Skeletal-Input> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities
	Add an Action and Controller Binding To use the Skeletal Input API, first add a "skeletal action" entry for each hand to the actions section of your application's action manifest file. Make sure that the 'type' is skeleton, and that it has an additional 'skeleton' field to define which skeleton the action should use. The choices for skeleton are either "/skeleton/hand/left" or "/skeleton/hand/right". For example: <pre data-bbox="514 483 1022 613">{ "name": "/actions/demo/in/lefthand_anim", "type": "skeleton", "skeleton": "/skeleton/hand/left" }</pre> Next, you need to add a binding for this action to each controller. You can do this through the SteamVR binding UI (preferred). The <i>Half-Life: Alyx</i> installation includes json files under “Half-Life Alyx/games/hlvr/cfg” that confirm this configuration. For example, the action manifest file, “actions.json,” includes the following skeletal action entries: <pre data-bbox="514 901 1438 1182">215 { 216 "name" : "/actions/default/in/HandSkeletonLeft", 217 "skeleton" : "/skeleton/hand/left", 218 "type" : "skeleton" 219 }, 220 { 221 "name" : "/actions/default/in/HandSkeletonRight", 222 "skeleton" : "/skeleton/hand/right", 223 "type" : "skeleton" 224 },</pre> The Knuckles bindings file, “bindings_knuckles.json,” includes the bindings of these actions:

Claim 15	Valve's Accused Instrumentalities
	<pre> 26 "skeleton" : [27 { 28 "output" : "/actions/default/in/handskeletonleft", 29 "path" : "/user/hand/left/input/skeleton/left" 30 }, 31 { 32 "output" : "/actions/default/in/handskeletonright", 33 "path" : "/user/hand/right/input/skeleton/right" 34 } 35], </pre> The bindings file includes additional bindings for the actions for each individual finger curl: <pre> 100 { 101 "inputs" : { 102 "pull" : { 103 "output" : "/actions/default/in/fingercurlindex" 104 } 105 }, 106 "mode" : "trigger", 107 "path" : "/user/hand/right/input/finger/index" 108 }, 109 { 110 "inputs" : { 111 "pull" : { 112 "output" : "/actions/default/in/fingercurlmiddle" 113 } 114 }, 115 "mode" : "trigger", 116 "path" : "/user/hand/right/input/finger/middle" 117 }, 118 { 119 "inputs" : { 120 "pull" : { 121 "output" : "/actions/default/in/fingercurlring" 122 } 123 }, 124 "mode" : "trigger", 125 "path" : "/user/hand/right/input/finger/ring" 126 }, </pre>

Claim 15	Valve's Accused Instrumentalities
	Further, the figure below is a screenshot taken from a Valve Index while testing this functionality. As just one example, curling an index finger towards the Index Controller's trigger causes the virtual finger to also curl, indicating that Valve's Accused VR Instrumentalities detect, by a proximity sensor, a non-contact interaction with a touch-sensitive device:  As another example in <i>Vertigo 2</i>, curling an index finger towards the Index Controller's trigger causes the virtual finger to also curl, indicating that Valve's Accused VR Instrumentalities detect, by a proximity sensor, a non-contact interaction with a touch-sensitive device:

Claim 15	Valve's Accused Instrumentalities
	
15[B] detecting, by a touch sensor, a touch on the touch-sensitive device;	Valve's Accused VR Instrumentalities detect, by a touch sensor, a touch on the touch-sensitive device. For example, Valve Index Controllers, also known as Knuckles,²⁶ use sensors to track pressure:²⁷

²⁶ See https://www.ifixit.com/Device/Valve_Index_Controller (last accessed May 10, 2023).

²⁷ See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities
	Sensor fusion Each controller uses 87 sensors to track hand position, finger position, motion, and pressure to determine user intent. All of these signals, combined with fine-tuned software and algorithms, give us a better understanding of how a player is holding and using the controllers. Further, the figure below is a screenshot taken from a Valve Index while testing this functionality. As just one example, touching the Index Controller's trigger causes the virtual finger to partially ball into the virtual hand, indicating that Valve's Accused VR Instrumentalities detect a touch on the touch-sensitive device:

Claim 15	Valve's Accused Instrumentalities
	 Note that the above hand pose differs from when a user pulls the trigger, which causes the finger to fully ball into the virtual hand, further indicating that Valve's Accused VR Instrumentalities detect a touch on the touch-sensitive device, as distinct from a trigger pull:

Claim 15	Valve's Accused Instrumentalities
	 As another example in <i>Vertigo 2</i>, touching the Index Controller's trigger causes the virtual finger to partially ball into the virtual hand, indicating that Valve's Accused VR Instrumentalities detect a touch on the touch-sensitive device:

Claim 15	Valve's Accused Instrumentalities
	
15[C] transmitting a first sensor signal associated with the non-contact interaction to a processor;	Valve's Accused VR Instrumentalities transmit a first sensor signal associated with the non-contact interaction to a processor. For example, Valve specifies certain minimum system specifications, including processor requirements:²⁸

²⁸ See <https://help.steampowered.com/en/faqs/view/105E-66E3-962A-1577> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities
	Minimum • OS: Windows 10, SteamOS, Linux • Processor: Dual Core with Hyper-Threading • Memory: 8 GB RAM • Graphics: Nvidia GeForce GTX 970 / AMD RX 480 • Network: Broadband Internet connection • Additional Notes: Available DisplayPort (Version1.2) and USB (2.0+) Port required Further, the APIs identified for Limitation 15A run on this processor, indicating that Valve's Accused VR Instrumentalities transmit a first sensor signal associated with the non-contact interaction to a processor.
15[D] transmitting a second sensor signal associated with the touch to a processor;	Valve's Accused VR Instrumentalities transmit a second sensor signal associated with the touch to a processor. For example, Valve specifies certain minimum system specifications, including processor requirements:²⁹ Minimum • OS: Windows 10, SteamOS, Linux • Processor: Dual Core with Hyper-Threading • Memory: 8 GB RAM • Graphics: Nvidia GeForce GTX 970 / AMD RX 480 • Network: Broadband Internet connection • Additional Notes: Available DisplayPort (Version1.2) and USB (2.0+) Port required Further, the APIs identified for Limitation 15B run on this processor, indicating that Valve's Accused VR Instrumentalities transmit a second sensor signal associated with the touch to a processor.
15[E] receiving, by the processor, the	Valve's Accused VR Instrumentalities receive, by the processor, the first and second sensor signals.

²⁹ See <https://help.steampowered.com/en/faqs/view/105E-66E3-962A-1577> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities								
first and second sensor signals;	For example, Valve specifies certain minimum system specifications, including processor requirements:³⁰ Minimum • OS: Windows 10, SteamOS, Linux • Processor: Dual Core with Hyper-Threading • Memory: 8 GB RAM • Graphics: Nvidia GeForce GTX 970 / AMD RX 480 • Network: Broadband Internet connection • Additional Notes: Available DisplayPort (Version1.2) and USB (2.0+) Port required Testing confirms the functionality described in Limitations 15A and 15B, support for both functions indicate that Valve's Accused VR Instrumentalities receive, by the processor, the first and second sensor signals.								
15[F] generating, by the processor, a haptic output signal based at least in part on the first and second sensor signals;	Valve's Accused VR Instrumentalities generate, by the processor, a haptic output signal based at least in part on the first and second sensor signals. For example, the Valve Index Controllers include an HD LRA for Haptics:³¹ Tech specs <table><tr><th>Inputs</th><th>Ergonomics</th><th>Connections</th><th>Haptics</th></tr><tr><td>A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU</td><td>Adjustable strap with pivot, antimicrobial tech fabric</td><td>USB-C, 2.4GHz Wireless</td><td>HD LRA</td></tr></table> Further, the “bindings_knuckles.json” configuration file includes bindings for haptics actions:	Inputs	Ergonomics	Connections	Haptics	A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless	HD LRA
Inputs	Ergonomics	Connections	Haptics						
A Button, B Button, System Button, Trigger, Thumbstick, Track Button with Force Sensor, Grip Force Sensor, Finger Tracking, IMU	Adjustable strap with pivot, antimicrobial tech fabric	USB-C, 2.4GHz Wireless	HD LRA						

³⁰ See <https://help.steampowered.com/en/faqs/view/105E-66E3-962A-1577> (last accessed May 10, 2023).

³¹ See <https://www.valvesoftware.com/en/index/controllers> (last accessed May 10, 2023).

Claim 15	Valve's Accused Instrumentalities
	<pre data-bbox="499 266 1509 613"> 4 "bindings" : { 5 "/actions/default" : { 6 "haptics" : [7 { 8 "output" : "/actions/default/in/hapticvibration", 9 "path" : "/user/hand/left/output/haptic" 10 }, 11 { 12 "output" : "/actions/default/in/hapticvibration", 13 "path" : "/user/hand/right/output/haptic" 14 } 15], 16 } 17 }</pre> Further, the figures below are screenshots taken from a Valve Index while testing this functionality. As just one example, as mentioned in Limitation 15A, curling an index finger towards the trigger causes a virtual finger to curl, as shown in the left figure below, which are screenshots taken from a Valve Index while testing this functionality. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects occur. Additionally, as mentioned in Limitation 15B, touching the trigger also causes the virtual finger to curl, as shown in the right figure below. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects also occur:

Claim 15	Valve's Accused Instrumentalities
	 Note that pulling the trigger results in the virtual hand grabbing the object, further indicating that Valve's Accused VR Instrumentalities independently track proximity to, touching, and pulling the trigger. Based on this behavior and on information and belief, the haptic output signal is based at least in part on the first and second sensor signals:

Claim 15	Valve's Accused Instrumentalities
	 As another example in <i>Vertigo 2</i>, as mentioned in Limitation 15A, curling an index finger towards the trigger causes a virtual finger to curl, as shown in the left figure below, which are screenshots taken from a Valve Index while testing this functionality. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects occur. Additionally, as mentioned in Limitation 15B, touching the trigger also causes the virtual finger to curl, as shown in the right figure below. If the resulting curl causes the virtual finger to touch a virtual object, haptic effects also occur:

Claim 15	Valve's Accused Instrumentalities
	Based on this behavior and on information and belief, Valve's Accused VR Instrumentalities generate a haptic output signal based at least in part on the first and second sensor signals. The Accused Instrumentalities literally infringe claim 15. If Valve asserts that this limitation is not literally present in Valve's Accused VR Instrumentalities as set forth above, it is found in Valve's Accused VR Instrumentalities under the doctrine of equivalents because any differences between such claim elements and Valve's Accused VR Instrumentalities are insubstantial and/or the accused products perform substantially the same function, in substantially the same way to achieve substantially the same result as the corresponding claim elements. The claimed function is generating, by the processor, a haptic output signal based at least in part on the first and second sensor signals. The claimed way is to base the haptic output signal at least in part on the first and second sensor signals. The claimed result is an outputted haptic effect, based at least in part on the first and second sensor signals. The function in Valve's Accused VR Instrumentalities is to process both a signal from the proximity sensor and a signal from the touch sensor, and generate a haptic effect in response to the described use case. On information and belief, the way the Valve's Accused VR Instrumentalities achieve this function is by receiving both the proximity signal and the touch signal. The observed result is the same as the claimed result. Under the insubstantial differences test, a person of ordinary skill in the art would view the differences as insubstantial and not affecting the overall function or result.

Immersion's Preliminary Infringement Contentions (10/20/2023)

Claim 15	Valve's Accused Instrumentalities
15[G] transmitting, by the processor, the haptic output signal to a haptic output device; and	Valve's Accused VR Instrumentalities transmit, by the processor, the haptic output signal to a haptic output device. Additionally, testing confirms the haptic effect for the use cases in Limitation 15[F], which indicates that Valve's Accused VR Instrumentalities, transmit, by the processor, the haptic output signal to a haptic output device.
15[H] outputting, by the haptic output device, a haptic effect in response to the haptic output signal.	Valve's Accused VR Instrumentalities output, by the haptic output device, a haptic effect in response to the haptic output signal. Additionally, testing confirms the haptic effect for the use cases in Limitation 15[F], which demonstrate that Valve's Accused VR Instrumentalities output, by the haptic output device, a haptic effect in response to the haptic output signal.

Claim 19	Valve's Accused Instrumentalities
19[P] A computer-readable non-transitory medium encoded with executable program code, the computer-readable medium comprising:	To the extent the preamble is determined to be limiting, Valve's Accused VR Instrumentalities include a computer-readable non-transitory medium encoded with executable program code, the computer-readable medium meeting the limitations described below.
19[A] program code for detecting, by a proximity sensor, a non-contact interaction with a touch-sensitive device;	Valve's Accused VR Instrumentalities include program code for detecting, by a proximity sensor, a non-contact interaction with a touch-sensitive device. See Limitation 15[A].

Immersion's Preliminary Infringement Contentions (10/20/2023)

Claim 19	Valve's Accused Instrumentalities
19[B] program code for detecting, by a touch sensor, a touch on the touch-sensitive device;	Valve's Accused VR Instrumentalities include program code for detecting, by a touch sensor, a touch on the touch-sensitive device. See Limitation 15[B].
19[C] program code for transmitting a first sensor signal associated with the non-contact interaction to a processor;	Valve's Accused VR Instrumentalities include program code for transmitting a first sensor signal associated with the non-contact interaction to a processor. See Limitation 15[C].
19[D] program code for transmitting a second sensor signal associated with the touch to a processor;	Valve's Accused VR Instrumentalities include program code for transmitting a second sensor signal associated with the touch to a processor. See Limitation 15[D].
19[E] program code for receiving, by the processor, the first and second sensor signals;	Valve's Accused VR Instrumentalities include program code for receiving, by the processor, the first and second sensor signals. See Limitation 15[E].
19[F] program code for generating, by the processor, a haptic output signal based at least in part on the first and second sensor signals; and	Valve's Accused VR Instrumentalities include program code for generating, by the processor, a haptic output signal based at least in part on the first and second sensor signals. See Limitation 15[F].
19[G] program code for transmitting the	Valve's Accused VR Instrumentalities include program code for transmitting the haptic output signal to a haptic output device.

Immersion's Preliminary Infringement Contentions (10/20/2023)

Claim 19	Valve's Accused Instrumentalities
haptic output signal to a haptic output device.	See Limitation 15[G].